(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 148 749 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
24.10.2001 Bulletin 2001/43

(51) Int Cl.⁷: H04Q 7/32

(21) Application number: 00117659.3

(22) Date of filing: 16.08.2000

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 17.04.2000 JP 2000121067

(71) Applicant: Hitachi, Ltd.
Chiyoda-ku, Tokyo 101-8010 (JP)

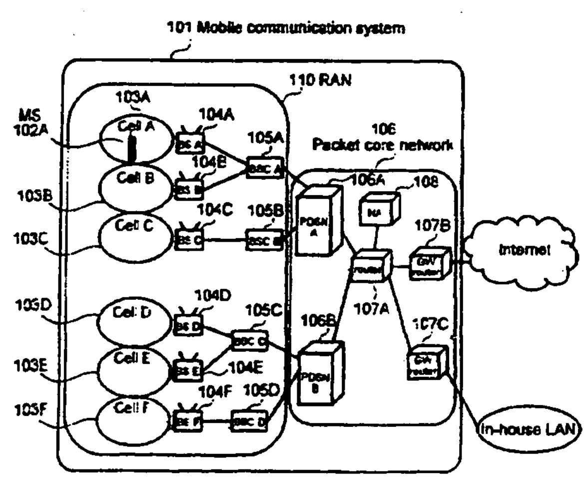
(72) Inventors:
- Mizutani, Mika, Hitachi Ltd., Int. Property Group
Chiyoda-ku, Tokyo 100-8220 (JP)
- Matsui, Susumu, Hitachi Ltd., Int. Property Group
Chiyoda-ku, Tokyo 100-8220 (JP)
- Hirata, Tetsuhiko, Hitachi Ltd., Int. Prop. Group
Chiyoda-ku, Tokyo 100-8220 (JP)
- Yano, Masahi, Hitachi Ltd., Int. Property Group
Chiyoda-ku, Tokyo 100-8220 (JP)

(74) Representative: Strehl Schübel-Hopf & Partner
Maximilianstrasse 54
80538 München (DE)

(54) **Mobile packet communication with prioritised access**

(57) A mobile communication system with a packet switching function which enables sharing of radio resources among mobile stations, wherein a mobile station which has generated a request for communication quality assurance periodically sends a packet for requesting preferential use of a radio channel in order to prevent timeout of the state transition timer, timeout of which would cancel radio channel assignment to the mobile station and bring it into a dormant state if a certain period elapses without transmission or reception of a signal, so that it can remain in the active state and hold the radio channel continuously. In addition, when the mobile station requesting communication quality assurance moves from one cell to another or requests radio channel assignment, the base station controller controls the radio base station so that the mobile station can be assigned a radio channel by sending a priority request packet.

FIG.1



EP 1 148 749 A2

Printed by Jouve, 75001 PARIS (FR)

Description

BACKGROUND OF THE INVENTION

[0001] This invention relates generally to mobile communication systems, and particularly to a mobile communication methods for mobile stations, base station controllers and packet data service nodes.

[0002] Effective use of radio channels in mobile communication systems can be achieved using packet switching techniques to enable sharing of a radio channel among mobile stations. According to such techniques, when a request for signal transmission or reception is generated, each mobile station uses the assigned radio channel shared with other mobile stations to transmit or receive signals in the form of packets. Further, when there is no such request, the radio channel is freed to enable its use by another mobile station. Moreover, in such techniques, radio channel assignment to mobile stations in which no packet transmission or reception has taken place for a certain time period are canceled.

[0003] In conventional mobile communication systems, if a request for signal transmission or reception occurs in a mobile station canceled for radio channel assignment, it is necessary to begin with assignment of a radio channel to that mobile station. But there is a possibility that there is no free radio channel. In addition, in a mobile communication system, as a mobile station moves from one cell to another, the mobile station must be assigned a radio channel by the radio base station controlling the destination cell but again there is a possibility that there is no free radio channel. This may lead to problems in communication quality assurance for communications that require high reliability, such as electronic commerce.

[0004] What is needed are techniques for maintaining an assigned radio channel between mobile and non mobile units in a packet based mobile communications systems.

SUMMARY OF THE INVENTION

[0005] According to the present invention, techniques for maintaining an assigned radio channel between a mobile unit and a non mobile unit when preferential use of the radio channel is desired are provided. Embodiments according to the invention can maintain the radio channel regardless of the packet transmission or reception interval, and can further assign radio channels preferentially according to requests for radio channel assignment. Techniques according to the invention can be embodied in a mobile unit, such as a mobile station, cell phone, pager and the like, a non-mobile unit, such as a base station controller, and the like, or a packet data service node.

[0006] In a representative embodiment according to the present invention, a mobile station requesting preferential use of a radio channel periodically sends a priority request to a radio base station, which, upon receiving the priority request, periodically sends a reply to the priority-requesting mobile station. This means that signal reception and transmission take place periodically or at regular time intervals between the priority-requesting mobile station and the radio base station; therefore, if this interval is shorter than a time allowed before cancellation of radio channel assignment, the mobile station can keep being assigned the radio channel.

[0007] In addition, according to this invention, the base station controller which controls the radio base station has means to separately control preferred mobile stations using radio channels preferentially and other mobile stations, or non-preferred mobile stations, and also to control the non-preferred mobile stations in the order of length of time which has elapsed after their transmission to, or reception from, the radio base station of the last signal (radio channel non-use time). Thus, if mobile stations requesting preferential use of a radio channel request assignment of a radio channel and there is no free radio channel in the radio base station, the base station controller can release radio channel assignments from non-preferred mobile stations, in the descending order of length of their radio channel non-use time, and re-assign the released radio channels to the priority-requesting mobile stations.

[0008] Numerous benefits are achieved by way of the present invention over conventional techniques.

[0009] An object of this invention is the provision of a mobile station which has means to keep an assigned radio channel, when preferential use of a radio channel is required, regardless of the packet transmission or reception interval, and also to be assigned preferentially a radio channel when it requests radio channel assignment.

[0010] A further object of the invention is the provision of a base station controller having means to keep assigning a radio channel to a mobile station which has requested preferential use of a radio channel, regardless of the packet transmission or reception interval and, when a priority-requesting mobile station requests assignment of a radio channel, assign it a radio channel preferentially.

[0011] Another object of the invention is the provision of a packet data service node having means to enable preferential use of a radio channel by a mobile station which has requested preferential use of a radio channel.

[0012] A further object of the invention is the provision of a mobile communication method that can keep assigning a radio channel to a mobile station which has requested preferential use of a radio channel, regardless of the packet transmission or reception interval and, when a priority-requesting mobile station requests assignment of a radio channel, assign it a radio channel preferentially.

[0013] These and other benefits are described throughout the present specification. A further understanding of the nature and advantages of the invention

herein may be realized by reference to the remaining portions of the specification and the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014]

Fig. 1 shows an example of a mobile data communication system according to this invention.
Fig. 2 shows an example of logical connection mapping between mobile station and PDSN.
Fig. 3 shows an example of threshold when radio channel assignment is made to mobile station.
Fig. 4 shows an example of the structure of a mobile station.
Fig. 5 shows an example of the structure of a base station controller.
Fig. 6 shows an example of the PDSN structure.
Fig. 7 shows resource state transition in packet switching.
Fig. 8 is a flowchart showing a sequence for a mobile station to hold a radio channel.
Fig. 9 shows the relationship among the state transition timer, PPP keep alive timer and QoS key state.
Fig. 10 is a flowchart showing a sequence for a mobile station to be assigned a radio channel preferentially to continue to use the channel.
Fig. 11 is a flowchart showing a processing sequence in PDSN which has received a QoS request.
Fig. 12 shows an example of the structure of a mobile station information table in the memory cache of PDSN.
Fig. 13 shows an example of the structure of a link layer connection control table in the memory cache of BSC.
Fig. 14 shows an example of the structure of a channel code control table in the memory cache of BSC.
Fig. 15 is a flowchart showing a processing sequence for BSC which has accepted an instruction for priority processing.
Fig. 16 is a flowchart showing a processing sequence for radio channel assignment in BSC when the priority-requesting mobile station requests radio channel assignment.
Figs. 17-20 show structures of a example request packets.

DESCRIPTION OF THE SPECIFIC EMBODIMENTS

[0015] Effective use of radio channels in mobile communication systems can be achieved using packet switching techniques to enable sharing of a radio channel among mobile stations. According to such techniques, when a request for signal transmission or reception is generated, each mobile station uses the assigned radio channel shared with other mobile stations to transmit or receive signals in the form of packets. Further, when there is no such request, the radio channel is freed to enable its use by another mobile station. Moreover, in such techniques, radio channel assignment to mobile stations in which no packet transmission or reception has taken place for a certain time period are canceled. For further information about packet wireless communication, reference may be had to a publication by 3rd Generation Partnership Project 2 (3GPP2), entitled, "Stage 3 description of Ax interface rev.1 (3gpp2-ACO-19990927-0)," the entire contents of which are incorporated herein by reference for all purposes.

[0016] In conventional mobile communication systems, if a request for signal transmission or reception occurs in a mobile station canceled for radio channel assignment, it is necessary to begin with assignment of a radio channel to that mobile station but there is a possibility that there is no free, radio channel. In addition, in a mobile communication system, as a mobile station moves from one cell to another, the mobile station must be assigned a radio channel by the radio base station controlling the destination cell but again there is a possibility that there is no free, radio channel. This may lead to a serious problem in communication quality assurance for communications that require high reliability, such as electronic commerce.

[0017] Embodiments according to the invention provide mobile communication systems, methods and apparatus having the capability to maintain an assigned radio channel. Responsive to a request by a mobile station for preferential use of a radio channel, in which the mobile station periodically sends a priority request to a radio base station, which, upon receiving the priority request, periodically sends a reply to the priority-requesting mobile station. This mechanism provides that signal reception and transmission take place periodically or at regular time intervals between the priority-requesting mobile station and the radio base station; therefore, if this interval is shorter than a time allowed before cancellation of radio channel assignment, the mobile station can keep being assigned the radio channel.

[0018] In addition, according to this invention, the base station controller which controls the radio base station can separately control preferred mobile stations using radio channels preferentially and other mobile stations, or non-preferred mobile stations. Further, the base station controller can control the non-preferred mobile stations in the order of length of time which has elapsed after their transmission to, or reception from, the radio base station of the last signal (radio channel non-use time). Thus, if mobile stations are requesting preferential use of a radio channel request assignment of a radio channel and there is no free radio channel in the radio base station, the base station controller can cancel radio channel assignments to non-preferred mobile stations, in the descending order of length of their radio channel non-use time, for example, and re-assign the freed radio channels to the priority-requesting mobile stations.

[0019] Fig. 1 shows the structure of a mobile data communication system 101 according to this invention. The system comprises of a radio access network (hereinafter called RAN) 110 and a packet core network 108, where the RAN, comprises of mobile stations (hereinafter called MS) 102, base stations (hereinafter called BS) 104 (104A-104F) which exchange signals with MS 102, located in service areas called cells 103 (103A-103F); and base station controllers (hereinafter called BSC) 105 (105A-105D) which comprehensively control the base stations 104, while the packet core network 109 comprises of packet data service nodes (hereinafter called PDSN) 106 (106A, 106B), which are connected with the radio access network 110 and have an IP packet routing function; a home agent (hereinafter called HA) 108 which enables mobile stations to move between PDSN 106A and 106B; gateway routers 107 (107B and 107C) for connection with external networks such as the Internet and an in-house LAN; and a router 107A which connects said gateway routers and PDSN 106.

[0020] Fig. 2 shows an example of mapping of connections between MS 102 and PDSN 106. A radio channel 203 is set between MS 102 and BSC 105 and a link layer connection 202 is set between BSC 105 and PDSN 106 so that PPP connection 201 is mapped on both the connections. Base station controller 105 controls the change in mapping of radio channel 203 and link layer connection 202 which occurs as a mobile station moves from one BS 104 to another BS 104, while PDSN 106 controls the change in mapping of PPP connection 201 and link layer connection 202 which occurs as a mobile station moves from one BSC 105 to another BSC 105.

[0021] Fig. 4 shows an example of the structure of MS 102 according to this invention. Mobile station 102 comprises of an antenna 404; a transmission/reception processor 403, which performs encoding and decoding to transmit or receive data through the antenna; a user interface section 401; a control section 402 which controls the user interface, carries out protocol processing of data and interfaces with the transmission/reception processor; and a battery 415. The user interface section 401 is composed of a display 407, a switch section 416, a speaker 413 and a microphone 414. The switch section 416 contains a power switch to turn on and off the power, 408; dial keys for entry of numerals and characters, 409; a select key which executes dialing, enables the commencement of talking with incoming lines and starts data service, 410; scroll keys to scroll the display, 411; and a QoS key to request communication quality assurance in accordance with inputs made by the user or instructions from the control section 402 which depends on the service used by the user, 412. The control section 402 comprises of a CPU 418, a ROM 406 and a RAM 405, where the CPU 418 starts the service depending on the request input from the switch section 416, performs transmission/reception traffic protocol processing related to the service and controls the display, the ROM406 stores the programs concerned and the RAM405 stores state information necessary for protocol processing and radio resource state information. A bus 417 interconnects 401, 402 and 403 in order to allow them to exchange data and programs.

[0022] Fig. 5 shows a representative structure for an example Base station controller, such as BSC 105, in a particular embodiment according to the present invention. Base station controller 105 comprises of a control section 501, a base station I/F port section 510 and a network I/F section 511. These sections are interconnected with each other through a packet bus 509. The control section 501 comprises of the following: a processor 503, which controls radio resources for each BS 104 and executes conversion between link layer connection 202 and radio channel 203; a memory 502 which stores the programs concerned; a memory cache 504 which contains tables to control radio channel codes as radio channel identifiers and tables for each MS to control radio channel information and radio resource states, a buffer memory 505 which temporarily stores data to be transmitted; a buffer memory controller 506; a hard disk 507; and a hard disk controller 508. The control section 501 is connected to base station 104 (four base stations in this embodiment) through the base station I/F port section 510. Further, control section 501 is connected to PDSN106 through the network I/F section 511.

[0023] Fig. 6 shows a representative structure for an example PDSN in a particular embodiment according to the present invention. Packet data service node 106A comprises a control section 601 and two or more routing sections 602, which are interconnected through a packet bus 603. The control section 601 comprises of the following: a memory 605A which stores a program to create packet routing tables; a processor 610A which executes that program; a memory cache 611A which contains packet routing tables and information about mobile stations, for example; a buffer memory 606A which stores packets, 505; a buffer memory controller 607A which comprises a function for DMA transfer of packets to and from the buffer memory 606A of the routing section 602, and a function of packet bus control; a hard disk controller 608; and a hard disk 609.

[0024] The packet routing table created by the processor 610A is used to control mobile IP processing; (including processes of collecting position information for the Mobile Stations 102 present in the mobile data communication system 101 and notifying Home Agent 108), establishing a PPP connection 201 with MS 102, establishing a link layer connection 202 with BSC 105, associating a mobile IP tunneling and PPP connection 201, and associating a PPP connection 201 and link layer connection 202.

[0025] The routing section 602 comprises a processor 610B which executes packet transmission between HA 108 and BSC 105 according to the packet routing table created by the control section. Further, routing section 602 can include a memory 605B; a buffer memory 606B; a buffer memory controller 607B; a memory cache 611B

which contains the packet routing table created by the control section, a port control section 612 which connects another router 107; and an internal bus. In this figure, one port control section 612 supports four ports and, in this embodiment, connections with more than one router 107 and more than one BSC 105 are made through these ports.

[0026] Fig. 7 shows a representative radio resource state transition diagram of packet switching in a particular embodiment according to the present invention. Fig. 7 illustrates three states: a null state 701, in which MS 102 is not connected to the mobile data communication system 101 (the power is off or data communication is impossible); an active state 702, in which MS 102 is connected to the mobile data communication system 101 and is assigned a radio channel; and a dormant state 703, in which MS 102 is connected to the mobile data communication system 101 but is not assigned a radio channel. With MS 102 in its active state 702, if a certain time has elapsed without transmission or reception of a signal, the radio channel assignment to the Mobile Station is canceled and the state shifts into the dormant state 703. In this specific embodiment of a mobile data communication system, mobile stations in the active state 702 can exchange packets with BS 104, while, in order to transmit or receive packets, mobile stations in the null state 701 or dormant state 703 request BSC 105 to assign them radio channels using random access channels or control channels. Upon having been assigned radio channels under the control of BSC 105, the mobile station can shift into the active state 702. A MS 102 which has failed to shift into the active state can request radio channel assignment again after a certain period has elapsed.

[0027] To enable state transitions as shown in Fig. 7, the control section 501 of BSC 105 is provided with a state transition timer 901B for each MS 102 so that when BSC 105 transmits a signal to, or receives a signal from, a MS 102, it restarts the state transition timer 901B corresponding to that Mobile Station. This process is illustrated graphically in Fig. 9. When this timer times out (a preset time expires), BSC 105 releases the radio channel from the corresponding MS 102, which then shifts from the active state 702 to the dormant state 703. In conventional mobile data communication systems, even if a mobile station is making communications which require high reliability, such as real-time applications and electronic commerce, if a certain time period has elapsed without any signal transmission or reception, resource state transition into the dormant state, i. e., state 703, occurs and the radio channel is released. Furthermore, since free radio channels are not always available, there may be a case in which the communication service concerned will become unavailable.

[0028] To overcome problems inherent to conventional technologies, when preferential use of a radio channel is needed, embodiments according to the present invention can inhibit state transitions, such as from the active state 702 to the dormant state 703, responsive to the user pushing the QoS key 412 of MS 102 or the control section 402 of MS 102 giving an instruction for preferential use of a radio channel, for example. Depending on the service in use by the user, in specific embodiments, the MS 102 can hold the radio channel continuously.

[0029] Fig. 8 shows a flowchart of a representative processing of a priority request input sequence in a particular embodiment according to the present invention. The processing illustrated by Fig. 8 can take place in MS 102, for example, in order to continuously hold the radio channel assigned by BS 104. When the user inputs to the QoS key 412, or when the control section 402 gives an instruction for preferential use depending on the service or application in use by the user, then, as illustrated by a step 801, the MS 102 turns on the QoS key. In a step 802, MS 102 can transmit PPP keep alive packet at regular intervals in order to prevent BSC 105 from releasing the radio channel from the MS. Structure of the PPP keep alive packet 1800 is shown in Fig.18. The PPP keep alive packet has MS ID field 1801 and packet ID field 1802 indicating that the packet is a PPP keep alive packet. If the communication system adopts CDMA scheme and the network can recognize the MS from the spreading code used in the packet, the MS ID field 1801 is not necessary. Then, in a step 803, MS 102 sets the PPP keep alive timer 902, provided in its control section 402, for measuring PPP keep alive packet transmission intervals, to a value smaller than the value set on the wireless channel state timer 901A. Wireless channel state timer 901A is in the Control Unit 402 in the MS 102, and measures the period between the last transmission/reception to/from BSC 105 and the release of assigned wireless channel by BSC 105. Therefore the Wireless channel state timer 901A measures in the same way as the State transfer timer 901B as a result.

[0030] Fig. 9 shows an example of relationship among the wireless channel state timer 901A, the state transition timer 901B PPP keep alive timer 902 and QoS key state. With the QoS key 412 on, because the PPP keep alive timer 902 is set to a value smaller than the wireless channel state timer 901A, the PPP keep alive timer 902 times out before timeout of the state transition timer 901B which would cause the release of the radio channel from MS 102 and the change of the state of MS 102 from the active state 702 to the dormant state 703. When the PPP keep alive timer 902 has timed out but the wireless channel state timer 901A has not timed out yet. MS 102 sends a PPP keep alive packet. As BSC 105 receives the PPP keep alive packet, it sends an acknowledgement packet to MS 102. Structure of the acknowledgement packet 1900 for the PPP keep alive packet is shown in Fig.19. The PPP keep alive packet has MS ID field 1901 and packet ID field 1902 indicating that the packet is an acknowledgement packet. If the communication system adopts CDMA scheme and the network

can recognize the MS from the spreading code used in the packet, the MS ID field 1902 is not necessary. Mobile station 102 restarts the wireless channel state timer 901A upon transmitting the PPP keep alive packet or receiving the acknowledgement packet from BSC 105, while BSC 105 restarts the state transition timer 901B upon receiving the PPP keep alive packet from MS 102 or transmitting the acknowledgement packet to MS 102, so that release of radio channel can be avoided.

[0031] When the user inputs to the QoS key 412 again at the end of use of service or when the control section 402 gives an instruction for cancellation of preferential use of the radio channel at the end of use of service, the QoS key is turned off in a step 804. Then, in a step 805, the PPP keep alive timer 902 is set to a normal value, or a value larger than the one set on the wireless channel state timer 901A. If a certain period has elapsed without any packet transmission or reception, the state transition timer 901B times out earlier than the PPP keep alive timer 902, the wireless channel is release from the MS 102, and the state of the MS 102 transfers into the dormant state 703. After the transition into the dormant state, when the PPP keep alive timer 902 times out, MS 102 does not send a PPP keep alive packet as long as the wireless channel state timer 901A is still time out.

[0032] When MS 102 shifts from its dormant state into its active state, or when MS 102 moves from one cell to another, it requests radio channel assignment from BSC 105 by the use of a random access channel or control channel. Fig. 17 shows a representative composition of a radio channel assignment request packet as an example. Here, 1701 represents the MS ID number that is requesting channel assignment and 1702 represents the BS ID number to which the MS is requesting the channel assignment. A field 1703 represents the transmission power level in a perch channel (hereinafter called BCCH) through which BS 104 is transmitting signals, while field 1704 represents the interference level in the uplink channel. A field 1705 denotes the received power of BCCH measured in MS 102 and 1706 the received SIR of BCCH. A field 1707 denotes the requested transmission speed of the downlink channel and 1708 that of the uplink channel. Some specific embodiments may comprise other informational fields, or may omit one or more of the fields illustrated in Fig. 17 without departing from the scope of the claimed invention. When MS 102 moves from one cell to another, BSC 105 can automatically catch the radio channel assignment request due to that movement without part or all of the information shown in Fig. 17, because BSC 105 knows the service and the transmission speed of channels used by that MS.

[0033] In a representative embodiment according to the present invention, base station controllers periodically collect from BS 104 communication quality information for each cell, such as desired signal level (RSSI), interference signal level (ISSI), desired-to-undesired signal ratio (SIR) and frame error rate (FER), and stores it in the memory 502. As BSC 105 receives the request for radio channel assignment from MS 102, it decides whether to assign a radio channel to that MS 102, depending on whether the predicted interference level is within a predetermined allowable range. The information on the cell in the memory 502, as well as information included in the radio channel assignment request packet, such as the requested transmission speed, SIR of BCCH and uplink channel interference level, are used for the processor 503 to predict how much the communication quality will deteriorate if a radio channel is assigned to the assignment requesting mobile station. Alternatively, the BSC105 can specifically predict what the interference level will be if radio channel assignment to the requesting mobile station takes place. Also, in specific embodiments, instead of using an interference level, the base station controller may decide whether to assign a radio channel depending on whether the transmission speed total for all active mobile stations connected to the base station to which the MS is requesting radio channel assignment, exceeds a preset threshold. For further description of communication quality information, reference may be had to a publication entitled, "ARIB STD-T53, a standard for CDMA portable mobile telephone systems established by the Association of Radio Industries and Businesses (ARIB)," the entire contents of which are incorporated herein by reference for all purposes.

[0034] Fig. 10 shows a flowchart of a representative priority request input sequence in a particular embodiment according to the present invention. The MS 102 can use such a sequence to enable the user requesting preferential use of a radio channel to be assigned a radio channel preferentially and to be able to use the assigned channel continuously. When the user inputs to the QoS key 412, or when the control section 402 gives an instruction for preferential use of a radio channel, depending on the service selected by the user, then, in a step 1001, MS 102 transmits a QoS requesting packet to the PDSN106 connected to it. If the prior MS 102 moves between cells and the BSC 105 has already recognized the MS as the MS requesting preferential channel assignment, the QoS requesting packet transmission is not necessary for that prior MS. Structure of the QoS requesting packet 2000 is shown in Fig.20. The QoS requesting packet has MS ID field 2001 and packet ID field 2002 indicating that the packet is a QoS requesting packet. If the communication system adopts CDMA scheme and the network can recognize the MS from the spreading code used in the packet, the MS ID field 2001 is not necessary. Having received the QoS requesting packet and decided whether or not to permit preferential channel management for MS 102, the PDSN 106 transmits a reply which is awaited by MS 102 in a step 1002. If, in step 1003, the reply is determined to be affirmative, then, in a step 1004, the QoS key is turned on and in a step 1005, a request for radio channel assignment is sent to BSC 105. When the radio channel assignment

request is made due to movement of MS 102 from one cell to another, BSC 105 may automatically knows the radio channel assignment request. If BSC 105 does not permit radio channel assignment, such request may be issued again after a certain period of time has elapsed. If BSC 105 permits radio channel assignment, as described herein with reference to Fig. 8, then in a step 1006, the MS 102 sends a PPP keep alive packet at regular intervals in order to hold the PPP connection 201. Then, in a step 1007, the PPP keep alive timer 902 is set to a value smaller than the wireless channel state timer 901A. With this setting, MS 102 can hold its active state 702. If the subscription contract for MS 102 prevents PDSN 106 from permitting preferential channel management, then, in a step 1008, the display of that mobile station MS 102 shows that the QoS function is invalid. If that is the case, then in a step 1009, the MS 102 requests radio channel assignment from BSC 105, as an ordinary mobile station, or a mobile station which is not preferentially controlled. If it is not assigned a radio channel, it may make the same request again after a certain period of time. If it is assigned a radio channel, the MS 102 does not send a PPP keep alive packet because it is not subject to preferential control. When the user inputs to the QoS key 412 again at the end of use of service, or when the control section 402 gives an instruction for cancellation of preferential use of the radio channel at the end of use of service, then in a step 804, the QoS key is turned off. Then, in a step 805, the PPP keep alive timer 902 is set to a normal value, or a value larger than the one set on the wireless channel state timer 901A.

[0035] Fig. 11 shows a flowchart of representative processing by a packet data service node responsive to a QoS request from a mobile station in a particular embodiment according to the present invention. In Fig. 11, PDSN 106 has received a QoS request from MS 102. Fig. 11 illustrates processing that is carried out at the processor 610A in the control section 601 of PDSN 106. After receiving the QoS request, PDSN 106 searches the mobile station information table 1201 corresponding to the requesting mobile station in a step 1101. Fig. 12 shows a representative structure of a mobile station information table 1201 as an example. The mobile station information table is located in the memory cache 611A of PDSN 106. The table 1201 contains a mobile station unique identifier obtained from the subscriber information and a temporary mobile station identifier assigned after connection with the mobile communication network; authentication and confidential information; IP address in use by mobile station; positional information; home network identifier; home agent address; and QoS contract service information 1202 comprising of information on existence of a priority processing contract 1203 and contract transfer throughput 1204. Some specific embodiments may comprise other informational fields, or may omit one or more of the fields illustrated in Fig. 12 without departing from the scope of the claimed invention.

[0036] After searching the mobile station information table, in a step 1102, the PDSN 106 checks the QoS service information 1202 to see if the mobile station is under the contract for priority processing. If the mobile station is not, then, in a step 1106, it informs the MS 102 that preferential control is unavailable. On the other hand, if it is under the contract for priority processing, then in a step 1103, the PDSN 106 gives an instruction for priority processing of the MS 102 to the BSC 105 connected to the MS 102. In a step 1104, the BSC 105 returns a reply for confirmation, and in a step 1105, notifies the mobile station that it can be preferentially controlled.

[0037] Base station controller 105 is provided with a link layer connection control table 1301 for each mobile station in order to control mapping of link layer connection 202 and the radio channel 203 assigned to the MS 102. Fig. 13 shows a representative structure of a link layer connection control table 1301 as an example. Located in the memory cache 504 of BSC 105, the link layer connection control table 1301 comprises a link layer connection identifier; mobile station IP address; resource state information 1302; uplink channel code and downlink channel code to identify the radio channel 303; packet escape queue; presence or absence of priority request 1303; uplink channel transmission speed 1304; downlink channel transmission speed 1305; uplink channel SIR 1306; downlink channel SIR 1307; and a control pointer. Some specific embodiments may comprise other informational fields, or may omit one or more of the fields illustrated in Fig. 13 without departing from the scope of the claimed invention.

[0038] The BSC 105 is also provided with a channel code control table 1401 for each of the cells 103 under the control of BS 104 in order to control the radio channel codes in use and enable preferential channel management. Fig. 14 shows a representative structure of a channel code control table 1401 as an example. The channel code control table 1401, located in the memory cache 504 of BSC 105, comprises of two queues: one is a preferred mobile station control queue 1402, which registers link layer connection control tables 1301 for the MS 102 under the preferential channel management. The other is a normal mobile station control queue 1403 which registers link layer connection control tables 1301 for the MS 102 under the priority processing contract but not under the preferential channel management, as well as the ones not under the priority processing contract. Each time MS 102 transmits or receives a signal through a radio channel, the link layer connection control table 1301 corresponding to that MS 102 is re-registered at the top of the control queue 1402 or 1403 by the processor 503 located in the control section 501 of BSC 105. Therefore, link layer connection control tables 1301 are registered in the control queues 1402 and 1403, from top to bottom thereof, in the ascending order of length of time which has elapsed after reception or

transmission of the final signal, or according to the rule that the table with the shortest non-use time is registered first and that with the longest non-use time is registered last.

[0039] Fig. 15 shows a flowchart of representative processing in a base station controller which has accepted the instruction for priority processing from a packet data service node in a particular embodiment according to the present invention. This processing is executed by the processor 503 located in the control section 501 of BSC 105 responsive to an instruction for priority processing from PDSN 106, for example. In Fig. 15, a mobile station MS 102 is already in its active state, and has transmitted QoS requesting packet to PDSN 106. The PDSN 106 has given BSC 105 an instruction for priority processing of the mobile station. In a step 1501, base station controller 105 searches the link layer connection control table 1301 corresponding to the MS 102 which should be processed preferentially. Then, in a step 1502, base station controller 105 turns on the priority request 1303 in the corresponding table. In a step 1503, this table is removed from the normal mobile station control queue 1403 in the channel code control table 1401 and re-registered at the top of the preferred mobile station control queue 1402. In a step 1504, the PDSN 106 is notified of completion of processing in response to the priority processing instruction.

[0040] Fig. 16 shows a flowchart of a representative control process for assigning a radio channel to a priority-requesting mobile station preferentially in a particular embodiment according to the present invention. This processing is executed by the processor 503 in the control section 501 of BSC 105, for example. In Fig. 16, a priority-requesting mobile station MS 102 is in its active state 702 and moves between base stations BS 104 (cells 103), or a priority-requesting mobile station MS 102 shifts from the null state 701 or dormant state 703 into the active state 702. When a mobile station MS 102 moves from one cell to another, this movement is detected in a step 1601. Then, in a step 1602, the link layer connection control table 1301 for that mobile station is removed from the channel code control table 1401 corresponding to the former base station BS 103. In a step 1603, utilizing the link layer connection control table as shown in Fig. 13, calculation is made for uplink and downlink channels separately to find the transmission speed total for all mobile stations in their active state under the control of the new BS 103 to which a radio channel assignment request is made; and then according to the radio channel assignment request packet received from the priority-requesting MS 102 or the transmission speed of the channel used by the MS 102 before its movement, it is judged whether or not a preset threshold will be exceeded if the transmission speed as requested by the priority-requesting MS 102 is assigned to the MS. Alternatively, judgment may be made as to whether the threshold as shown in Fig. 3 will be exceeded by the result of calculation from the transmission speed total for each of the uplink and downlink channels in case of radio channel assignment being made to the priority-requesting MS 102, as well as the interference signal level calculated by the processor 503. If the threshold is not to be exceeded, then, in a step 1604, radio channel assignment is made to the MS 102 and the link layer connection control table 1301 is registered at the top of the preferred mobile station control queue 1402 in the channel code control table 1401 corresponding to the new BS102 in a step 1605. Otherwise, if the threshold would be exceeded, in a step 1606, a judgment is made as to whether there is a link layer connection control table 1302 registered in the normal mobile station control queue 1403 in the channel code control table 1401 of the new BS 103. If there is no link layer connection control table 1301 registered in the normal mobile station control queue 1403, it is impossible to make radio channel assignment to the priority-requesting mobile station because all radio channels are in use by preferred mobile stations, as indicated by step 1612. If there is a link layer connection control table 1301 registered in the normal mobile station control queue 1403, then in a step 1607, a judgment is made as to whether, if radio channel assignments for normal or non-preferred mobile stations whose link layer connection tables are registered in the normal mobile station control queue 1403 are all canceled and radio channel assignment is made to the priority-requesting mobile station, the transmission speed total will exceed the threshold. Alternatively, judgment may be made as to whether, by calculating the interference level and transmission speed total for the case that radio channel assignment for all normal mobile stations are canceled and radio channel assignment is made to the priority-requesting mobile station, the interference level will exceed the threshold as shown in Fig. 3. If the transmission speed total or the interference level is to exceed the threshold, then it is impossible to make radio channel assignment to the priority-requesting mobile station, as indicated by step 1612. In any case other than the above, the normal mobile stations whose link layer connection control tables 1301 are registered in the normal mobile station control queue 1403 are canceled for radio channel assignments in reverse order of registration, or on the basis of first cancellation of last registered mobile station, until the transmission speed total for all mobile stations or the interference level comes below the threshold if radio channel assignment is made to the priority-requesting mobile station, thus forcing them to shift into the dormant state, indicated by steps 1608, 1609 and 1610. Then, in a step 1611, a radio channel freed from a normal mobile station or non-preferred MS is assigned to the priority-requesting mobile station and, in a step 1605, the link layer connection control table 1301 for the priority-requesting mobile station is registered at the top of the preferred mobile station control queue in the channel code control table corresponding to the base station which has made the assignment.

[0041] Thus, in representative embodiments according to the present invention, when the user or the application in use needs communication quality assurance, by having the mobile station concerned periodically send a packet to request preferential use of a radio channel, it is possible to prevent timeout of the state transition timer, which counts the timing of transition from the active state to the dormant state, so that the priority-requesting mobile station can hold the radio channel continuously.

[0042] Furthermore, the base station controller is provided with means to separately control, for each cell, radio channels used preferentially by preferred mobile stations and radio channels used by normal mobile stations, as well as means to control the radio channels in use by normal mobile stations in the order of length of time which has elapsed after transmission or reception of the last signal. When a priority-requesting mobile station moves from one cell to another, or when the priority-requesting mobile station requests radio channel assignment, if there is no free channel in the cell, a normal mobile station among the ones in the cell which have been assigned radio channels is forced to be canceled for the radio channel assignment, in the descending order of length of time which has elapsed after transmission or reception of the final signal, and the mobile station thus canceled is forced to shift from the active state into the dormant state, while the radio channel thus freed is assigned to the priority-requesting mobile station, which makes it possible that the priority-requesting mobile station can hold the radio channel preferentially as it moves to another cell, or can be assigned a radio channel preferentially when newly requesting radio channel assignment.

[0043] The preceding has been a description of the preferred embodiment of the invention. It will be appreciated that deviations and modifications can be made without departing from the scope of the invention, which is defined by the appended claims.

Claims

1. A wireless communication device for communicating with at least one other device, said wireless communication device comprising:

   a CPU,
   a memory, a first timer and a second timer,
   a bus, connecting said CPU, said memory, to a communications interface;
   wherein when said wireless communications device requests preferential use of a communication channel, said first timer is set with a timeout value less than said second timer, such that a periodic transmission of a priority request is made by said CPU via said communication interface at expiration of said first timer provided that said second timer has not expired.

2. A communication device for communicating with a control station, said device having a control section comprising:

   a CPU,
   a memory,
   a bus, connecting said CPU, said memory to a communications interface;
   wherein when said communication device requests preferential use of a communication channel, said channel having been assigned by said control station, said CPU periodically causes sending of a priority request to said control station via said communications interface.

3. A mobile station for wireless communication with a base station, said mobile station having a control section comprising:

   a CPU,
   a memory,
   a bus, connecting said CPU, said memory to a transmission/reception processor;
   wherein when said mobile station requests preferential use of a wireless communication channel, said wireless communication channel having been assigned by said base station, said CPU periodically causes sending of a priority request to said base station via said transmission/reception processor.

4. The mobile station of claim 3, further comprising:

   a PPP keep alive timer which begins counting from a time of any of a last signal transmission and a last signal reception;
   wherein upon timeout of said PPP keep alive timer, said control section causes sending of said base station said priority request, and restarts said PPP keep alive timer.

5. The mobile station as defined in claim 4, wherein,
   if said radio channel is used preferentially, said control section sets a counting period for said PPP keep alive timer to a value smaller than a channel holding period. said channel holding period being a time measured from any of a last signal transmission and a last signal reception, and until said base station cancels a radio channel assignment to said mobile station.

6. The mobile station as defined in claim 5, further comprising:

   a wireless channel state timer that counts said channel holding period; and wherein

if said wireless channel state timer reaches said channel holding period, said control section discontinues sending said priority request.

7. A mobile station for wireless communication with a base station, said mobile station having a control section comprising:

a CPU,
a memory,
a bus, connecting said CPU, said memory to a transmission/reception processor;
wherein when said mobile station requests assignment of a wireless channel from said base station, said CPU causes transmission of a preferential channel assignment request via said transmission/reception processor.

8. The mobile station of claim 7, wherein said preferential channel assignment request further comprises an identification of a preferential channel usage request packet.

9. A base station controller for controlling a base station, said base station communicating with at least one of a plurality of mobile stations, said base station controller comprising:

a base station interface, connecting said base station controller to said base station;
a control section;
a network interface, connecting said base station controller to a network;
a packet bus, interconnecting said base station interface, said network interface to said control section;
wherein said control section receives from at least one of said plurality of mobile stations a request to use a radio channel preferentially, through said base station, and responsive to said priority request, transmits an acknowledgement to said mobile station.

10. The base station controller of claim 9, said base station controller further comprising:
a state transition timer that counts a time period starting from at least one of a transmission to said base station and a reception from said base station, and until a channel assigned to said mobile station is released.

11. The base station controller of claim 9, wherein said acknowledgement further comprises identification of an acknowledgement packet for said priority request.

12. A base station controller for controlling a base station, said base station communicating with at least one of a plurality of mobile stations, said base station controller comprising:

a base station interface, connecting said base station controller to said base station;
a control section;
a network interface, connecting said base station controller to a network;
a packet bus, interconnecting said base station interface, said network interface to said control section;
wherein said control section receives from at least one of said plurality of mobile stations a request to use a radio channel preferentially, through said base station, and responsive to said priority request, determines an availability of a channel to assign to said mobile station.

13. The base station controller of claim 12, wherein responsive to a request for a priority wireless channel assignment by a priority requesting mobile station, and wherein, if there are no free wireless channels, said control section releases a wireless channel assignment from a non-priority requesting mobile station, and assigns said wireless channel so released to said priority requesting mobile station.

14. The base station controller of claim 13, wherein
said control section releases wireless channels from said non-priority requesting mobile station that has not transmitted nor received signals for a longest period of time.

15. The base station controller of claim 13, wherein when priority requesting mobile stations, moves from a control area of a first base station to a control area of a second base station, and requests a wireless channel assignment from said second base station, and wherein, if there are no free wireless channels, a control section of a base station controller that controls said second base station releases a wireless channel assignment from a non-priority requesting mobile station, and assigns said wireless channel so released to said priority requesting mobile station.

16. The base station controller of claim 13, further comprising a link layer connection control table for managing wireless channels used by each of said mobile stations, and

said link layer connection control table comprising a priority management registration field, wherein
said control section makes a priority management registration in said priority management registration field for said priority requesting mobile stations.

17. The base station controller of claim 16, further comprising:

a channel control table for registering said link layer connection control table for each of said base stations; said channel control table comprising:

a preferred mobile station control queue for registering said link layer connection control table of a preferred mobile station which receives said priority channel management from said base station controller, and

a non-preferred mobile station control queue for registering said link layer connection control table of said non-preferred mobile station in ascending order of length of time for which said non-preferred mobile station left said wireless channel unused.

18. The base station controller of claim 17, wherein said link layer connection control table includes a uplink transmission speed field, and a downlink transmission speed field, wherein

said control section calculates total transmission speed of all mobile stations in a cell controlled by said base station, and

in case of said total transmission speed being over a threshold, said control section releases said wireless channel from said non-preferred mobile station whose link layer connection control table is registered last in said non-preferred mobile station control queue.

19. The base station controller as defined in claim 17, wherein

said link layer connection control table includes a uplink SIR field, and a downlink SIR field,

said control section calculates total SIR in a cell controlled by said base station, and

in case of said total SIR being over a predetermined threshold said control section releases said wireless channel from said non-preferred mobile station whose link layer connection control table is registered at the last of said non-preferred mobile station control queue.

20. A packet data service node for connecting a base station controller with an external network, said packet data service node comprising:

a control section;

a routing section;

a bus, connecting said routing section to said control section

wherein. said control section, responsive to receiving through said base station controller a priority request from a priority-requesting mobile station, sends to said base station controller a reply to authorize priority processing, said reply enabling said priority-requesting mobile station to use a wireless channel preferentially.

21. The packet data service node of claim 20, further comprising:

a mobile station information table for registering information about priority channel usage authorization.

22. The packet data service node of claim 21, wherein:

said control section, upon receiving said priority request, refers to said mobile station information table and sends a reply to permit priority processing to said base station controller if said priority requesting mobile station has been registered for priority channel usage, otherwise sends a reply to disallow priority processing to said base station controller if said priority requesting mobile station has not been registered for priority channel usage.

23. A mobile communication system comprising:

a base station controlled by a base station controller;

at least one of a plurality of mobile stations, in communication with said base station using wireless communication channels, wherein

said base station controller releases a wireless communication channel from said mobile station when no transmission nor reception between said mobile station and said base station occurs for a time period exceeding a threshold period, and wherein:

a priority-requesting mobile station requesting a preferential use of a wireless communication channel sends a priority request to said base station; and

said base station controller receives said priority request through said radio base station, and upon reception of said priority request, said base station controller transmits acknowledgement to said mobile station.

24. The mobile communication system of claim 23, wherein:

said mobile station transmits said priority request periodically.

25. The mobile communication system of claim 23, wherein:

said base station controller does not release said wireless channel when a prior channel usage contract of said mobile station is registered to a packet data service node to which said base station

controller connected.

26. A mobile communication system comprising:

a base station controlled by a base station controller;
at least one of a plurality of mobile stations, in communication with said base station using wireless communication channels, wherein said base station controller releases a wireless communication channel from said mobile station when no transmission nor reception between said mobile station and said base station occurs for a time period exceeding a threshold period, and wherein:

a priority-requesting mobile station requesting a preferential use of a wireless communication channel sends a priority request to said base station; and
said base station controller receives said priority request through said radio base station, and thereupon assigns said wireless communication channel to said priority-requesting mobile station.

27. The mobile communication system comprising:

a radio base station;
a base station controller, operative to control said radio base station;
at least one mobile station; wherein said at least one mobile station is in communication with said radio base station; and wherein said base station controller receives a wireless channel assignment request from said mobile station, which transmitted a priority request for requesting priority use of a wireless channel, and wherein, in case of shortage of wireless channels exists, said base station controller releases a wireless channel from a non-preferred mobile station, and assigns said released wireless channel to said requesting mobile station.

28. The mobile communication system of claim 27, wherein said base station controller releases said wireless channel from said non-preferred mobile station which has not transmitted and/or received signals for a longest period.

29. The mobile communication system of claim 27, wherein said base station controller releases said wireless channel when a prior channel usage contract of said channel assignment requesting mobile station is registered to a packet data service node to which said base station controller connected.

30. The mobile communication system of claim 27, wherein said base station controller releases said wireless channel upon hand-off of said priority request transmitting mobile station.

31. A mobile station according to claim 3, wherein said priority request comprises identification of a preferential channel usage request packet.

32. A wireless communication method for establishing a priority communication channel, said wireless communication method comprising:

transmitting a priority request for use of a wireless channel;
setting a first timer with a timeout value less than a timeout value of a second timer;
periodically transmitting said priority request at expiration of said first timer provided that said second timer has not expired; and
wherein, responsive to said periodically transmitting said priority request, said priority use of said wireless channel is maintained.

33. A wireless communication method for establishing a priority communication channel, said wireless communication method comprising:

receiving a priority request for use of a wireless channel;
determining that a priority usage of said wireless channel is permissible;
setting a timeout value to a timer;
releasing said priority usage of said wireless channel at expiration of said timer unless any of: a periodic priority request is received, a transmission is received and a transmission is made; and
wherein, responsive to any of: a periodic priority request is received, a transmission is received and a transmission is made, said priority use of said wireless channel is maintained.

34. The wireless communication method of claim 33, further comprising:
transmitting an acknowledgement responsive to said priority request.

35. The wireless communication method of claim 33, wherein said determining that a priority usage of said wireless channel is permissible further comprises:
determining that said priority request is authorized under contract for priority usage.

36. A wireless communication method for establishing a priority communication channel, said wireless communication method comprising:

receiving a preferential channel assignment request;
determining that wireless channel resources are sufficient to fill said preferential channel assignment request;
if insufficient wireless channel resources exist to fill said preferential channel assignment request, releasing a wireless channel from another communication, and assigning said released wireless channel to satisfy said preferential channel assignment request.

37. The wireless communication method of claim 36, wherein said releasing a wireless channel from another communication further comprises:
releasing said wireless channel from a non-preferred communication which has not transmitted and/or received signals for a longest period.

38. The wireless communication method of claim 36, further comprising:
releasing said wireless channel upon a source of said preferential channel assignment request moving to a location serviced by a different cell.

39. A wireless communication apparatus for establishing a priority communication channel, said wireless communication apparatus comprising:

means for transmitting a priority request for use of a wireless channel;
means for setting a first timer with a timeout value less than a timeout value of a second timer;
means for periodically transmitting said priority request at expiration of said first timer provided that said second timer has not expired; and
wherein, responsive to said periodically transmitting said priority request, said priority use of said wireless channel is maintained.

40. A wireless communication apparatus for establishing a priority communication channel, said wireless communication apparatus comprising:

means for receiving a priority request for use of a wireless channel;
means for determining that a priority usage of said wireless channel is permissible;
means for setting a timeout value to a timer;
means for releasing said priority usage of said wireless channel at expiration of said timer unless any of: a periodic priority request is received, a transmission is received and a transmission is made; and
wherein, responsive to any of: a periodic priority request is received, a transmission is received and a transmission is made, said priority use of said wireless channel is maintained.

41. A wireless communication apparatus for establishing a priority communication channel, said wireless communication apparatus comprising:

means for receiving a preferential channel assignment request from a base station controller, and
means for transmitting a reply to authorize said base station controller to assign a wireless channel preferentially.

42. A computer program product for establishing a priority communication channel, said computer program product comprising:

code that transmits a priority request for use of a wireless channel;
code that sets a first timer with a timeout value less than a timeout value of a second timer;
code that periodically transmits said priority request at expiration of said first timer provided that said second timer has not expired; and
wherein, responsive to said periodically transmitting said priority request, said priority use of said wireless channel is maintained; and
a computer readable storage medium for holding the codes.

43. A computer program product for establishing a priority communication channel, said computer program product comprising:

code that receives a priority request for use of a wireless channel;
code that determines that a priority usage of said wireless channel is permissible;
code that sets a timeout value to a timer;
code that releases said priority usage of said wireless channel at expiration of said timer unless at least one of: a periodic priority request is received, a transmission is received and a transmission is made; and
wherein, responsive to at least one of: a periodic priority request is received, a transmission is received and a transmission is made, said priority use of said wireless channel is maintained; and
a computer readable storage medium for holding the codes.

44. A computer program product for establishing a priority communication channel, said computer program product comprising:

code that receives a preferential channel assignment request from a base station controller.

and
code that transmits a reply to authorize said base station controller to assign a wireless channel preferentially; and
a computer readable storage medium for holding the codes.

FIG.1

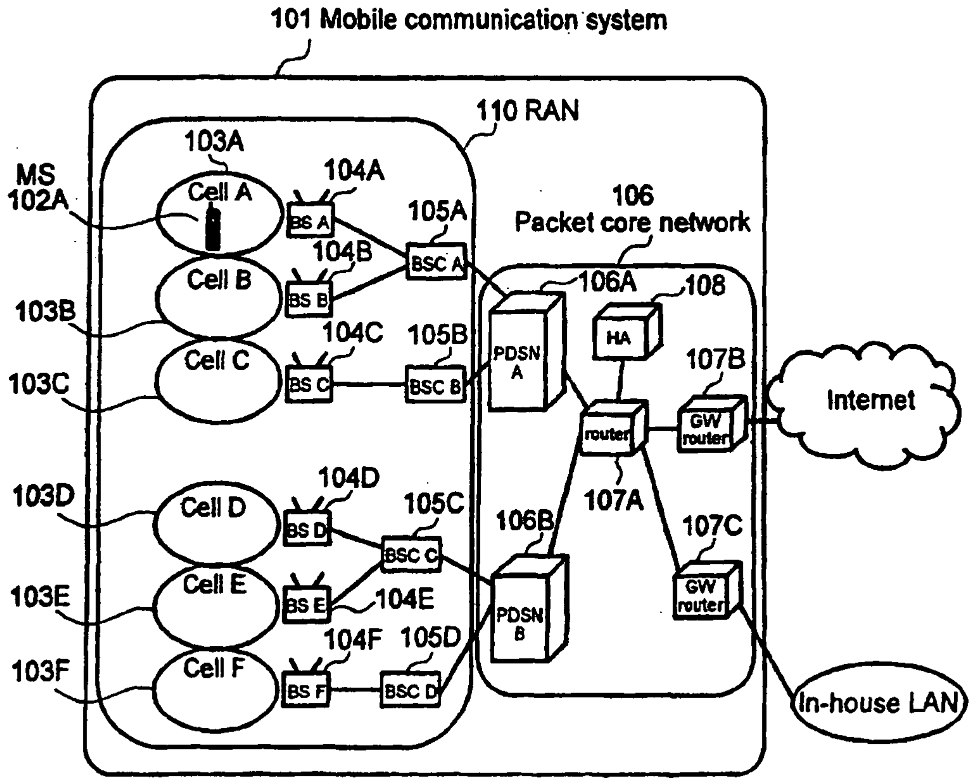

101 Mobile communication system
110 RAN
103A
MS
102A
Cell A
104A
BS A
105A
BSC A
104B
Cell B
BS B
103B
104C
105B
Cell C
BS C
BSC B
103C
106
Packet core network
106A
108
HA
PDSN
A
107B
router
GW
router
Internet
107A
103D
104D
Cell D
BS D
105C
BSC C
106B
107C
103E
Cell E
BS E
104E
GW
router
PDSN
B
104F
105D
103F
Cell F
BS F
BSC D
In-house LAN

FIG.2

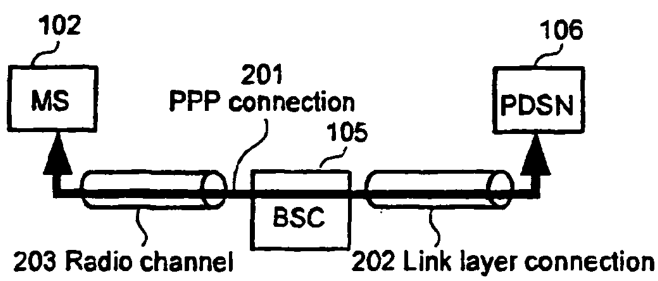

102
MS
201
PPP connection
106
PDSN
105
BSC
203 Radio channel
202 Link layer connection

FIG.3

300 302 302

Transmission speed total for all mobile stations (kbps)	Interference level threshold (dBm/Hz)
0	-165
64	-163
128	-161
256	-159
512	-157
1024	-155
2048	-153

FIG.4

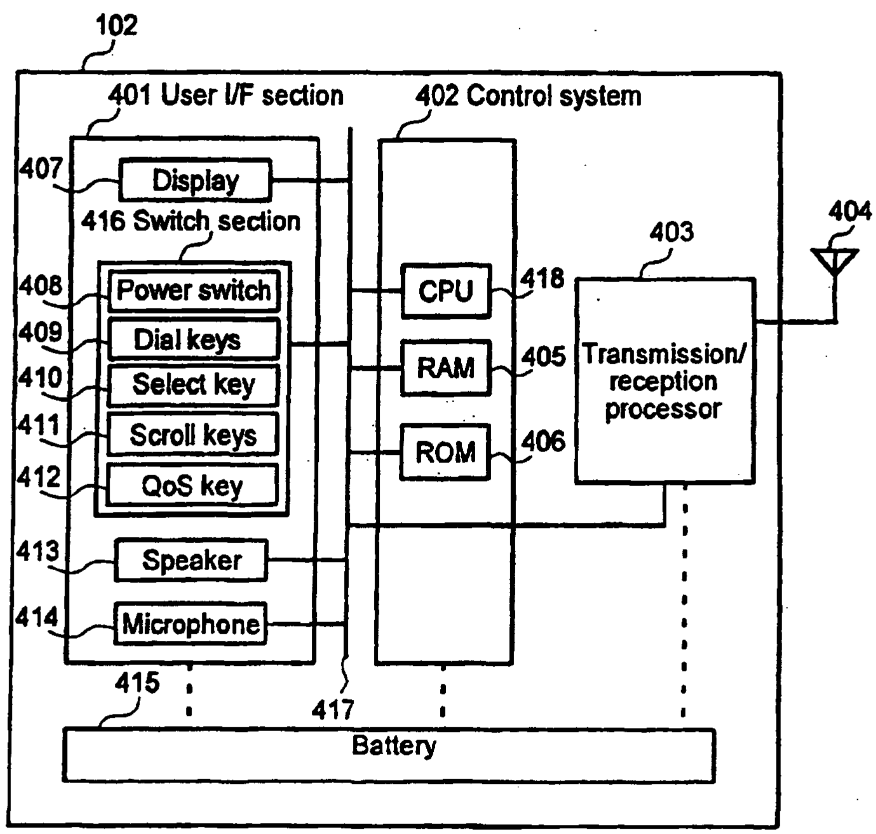

102
401 User I/F section
402 Control system
407
Display
416 Switch section
408
Power switch
409
Dial keys
410
Select key
411
Scroll keys
412
QoS key
413
Speaker
414
Microphone
415
417
CPU
418
RAM
405
ROM
406
403
Transmission/
reception
processor
404
Battery

FIG.5

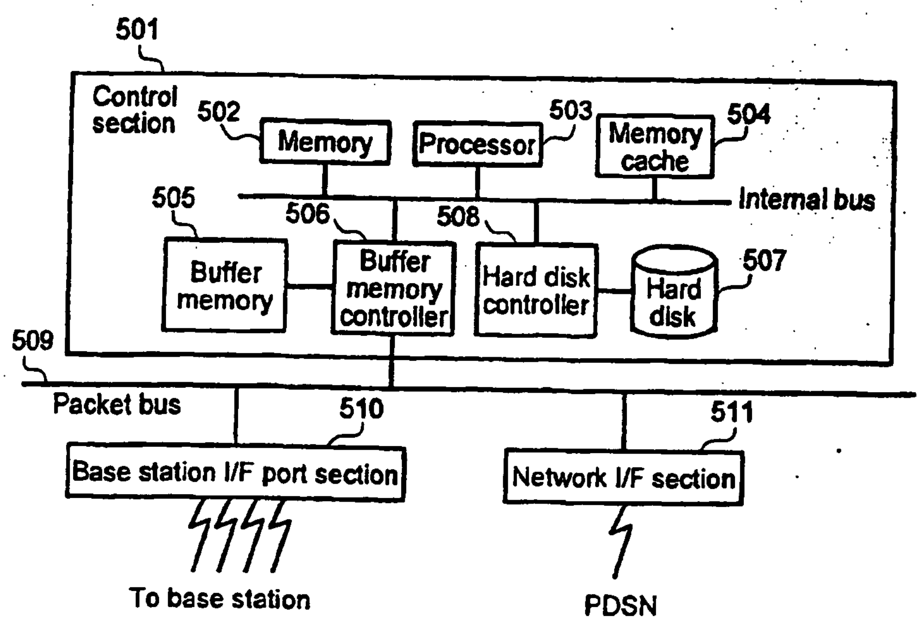

501
Control section
502
Memory
Processor
503
Memory cache
504
Internal bus
505
506
508
Buffer memory
Buffer memory controller
Hard disk controller
Hard disk
507
509
Packet bus
510
511
Base station I/F port section
Network I/F section
To base station
PDSN

FIG.6

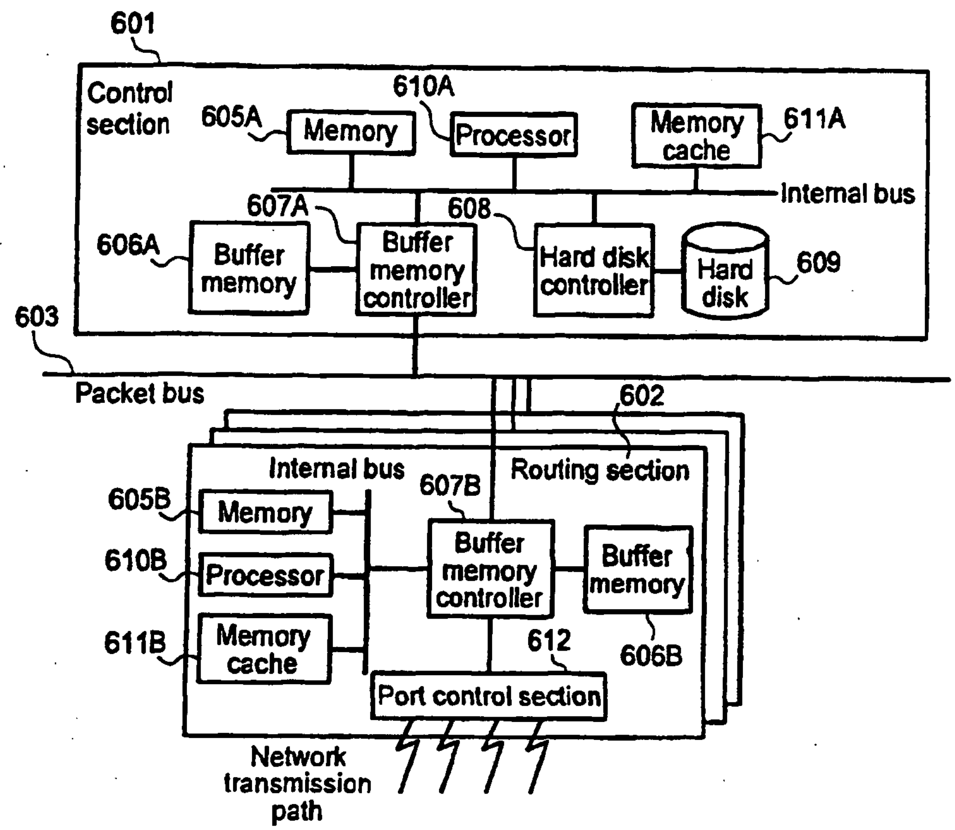

601
Control section
605A
Memory
610A
Processor
Memory cache
611A
Internal bus
607A
608
606A
Buffer memory
Buffer memory controller
Hard disk controller
Hard disk
609
603
Packet bus
602
Internal bus
607B
Routing section
605B
Memory
610B
Processor
Buffer memory controller
Buffer memory
611B
Memory cache
612
606B
Port control section
Network transmission path

FIG.7

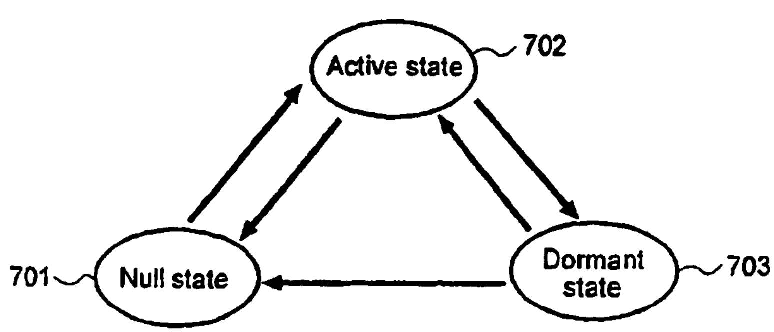

Active state
702
701
Null state
Dormant state
703

FIG.8

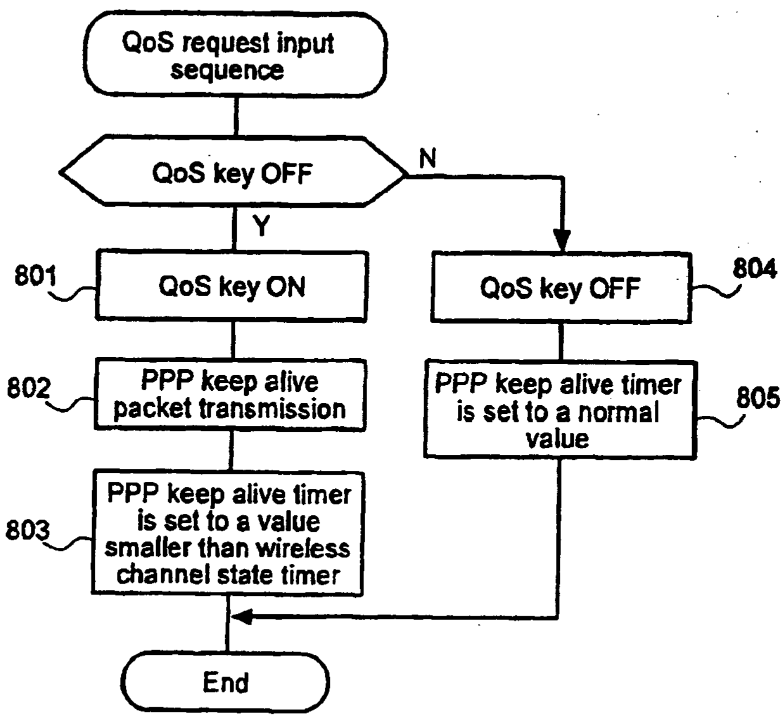

QoS request input sequence
QoS key OFF
N
Y
801
QoS key ON
804
QoS key OFF
802
PPP keep alive packet transmission
PPP keep alive timer is set to a normal value
805
803
PPP keep alive timer is set to a value smaller than wireless channel state timer
End

FIG.9

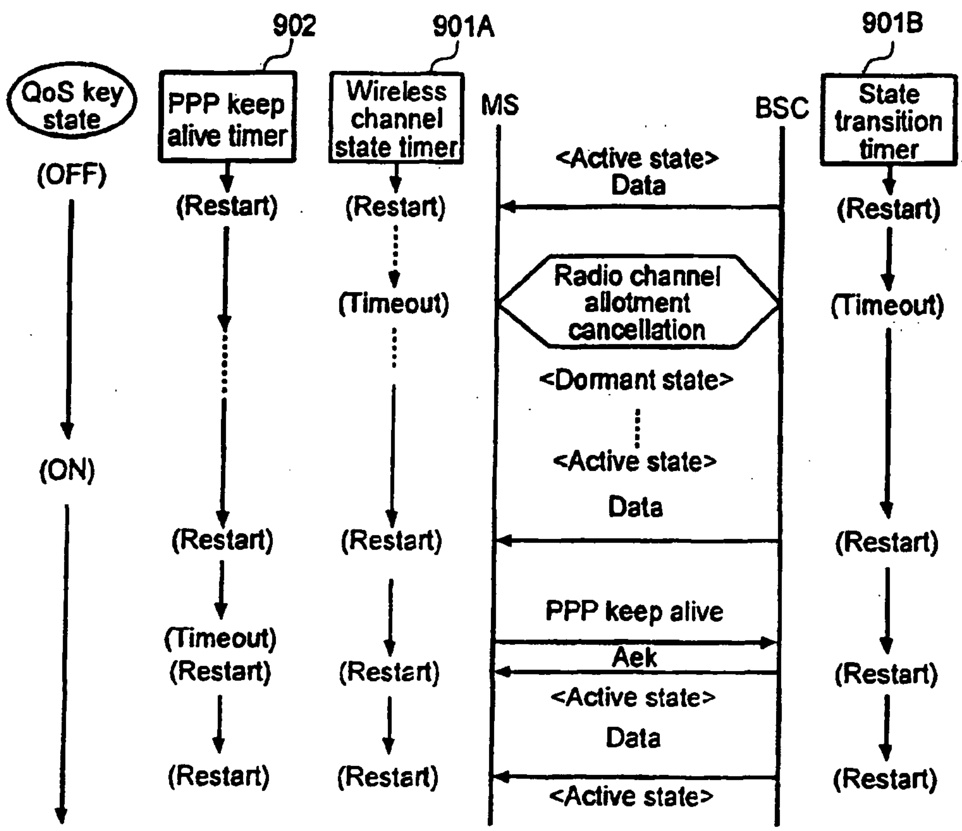

902
901A
901B
QoS key state
PPP keep alive timer
Wireless channel state timer
MS
BSC
State transition timer
(OFF)
<Active state>
Data
(Restart)
(Restart)
(Restart)
Radio channel allotment cancellation
(Timeout)
(Timeout)
<Dormant state>
(ON)
<Active state>
Data
(Restart)
(Restart)
(Restart)
PPP keep alive
(Timeout)
Aek
(Restart)
(Restart)
(Restart)
<Active state>
Data
(Restart)
(Restart)
(Restart)
<Active state>

FIG.10

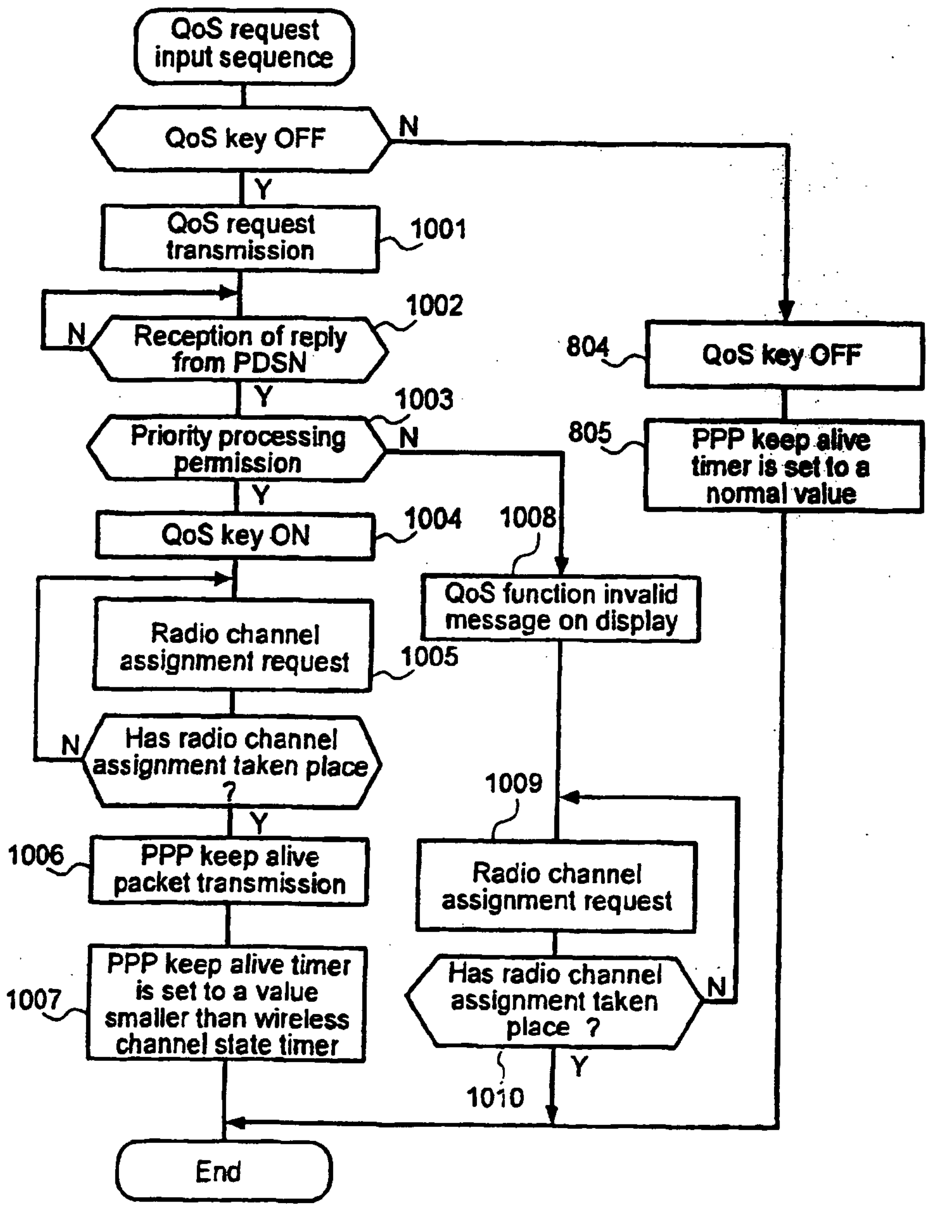

QoS request input sequence
QoS key OFF
N
Y
QoS request transmission
1001
1002
N
Reception of reply from PDSN
Y
1003
Priority processing permission
N
Y
QoS key ON
1004
1008
Radio channel assignment request
1005
QoS function invalid message on display
N
Has radio channel assignment taken place ?
Y
1006
PPP keep alive packet transmission
1007
PPP keep alive timer is set to a value smaller than wireless channel state timer
1009
Radio channel assignment request
Has radio channel assignment taken place ?
N
Y
1010
804
QoS key OFF
805
PPP keep alive timer is set to a normal value
End

FIG.11

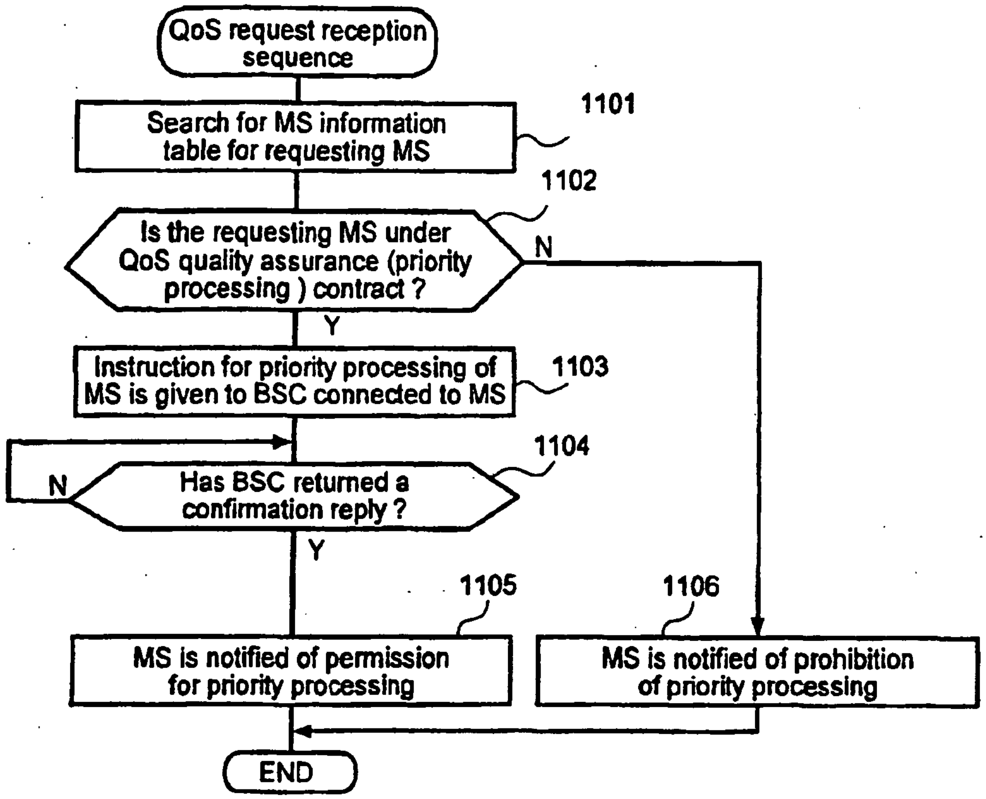

QoS request reception sequence
1101
Search for MS information table for requesting MS
1102
Is the requesting MS under QoS quality assurance (priority processing) contract ?
N
Y
Instruction for priority processing of MS is given to BSC connected to MS
1103
1104
Has BSC returned a confirmation reply ?
N
Y
1105
1106
MS is notified of permission for priority processing
MS is notified of prohibition of priority processing
END

FIG.12

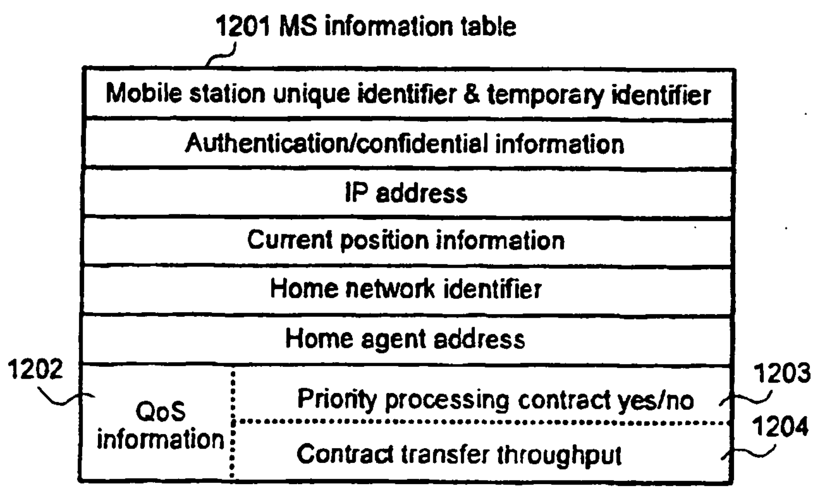

1201 MS information table
Mobile station unique identifier & temporary identifier
Authentication/confidential information
IP address
Current position information
Home network identifier
Home agent address
1202
QoS information
Priority processing contract yes/no
1203
Contract transfer throughput
1204

FIG.13

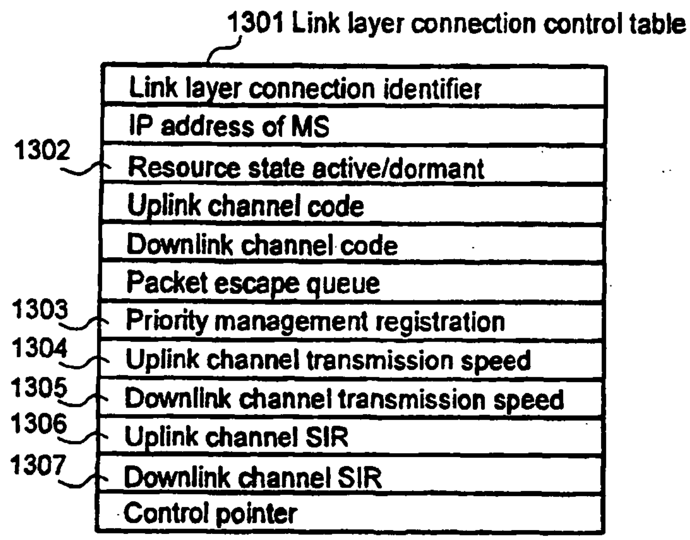

1301 Link layer connection control table
Link layer connection identifier
IP address of MS
1302
Resource state active/dormant
Uplink channel code
Downlink channel code
Packet escape queue
1303
Priority management registration
1304
Uplink channel transmission speed
1305
Downlink channel transmission speed
1306
Uplink channel SIR
1307
Downlink channel SIR
Control pointer

FIG.14

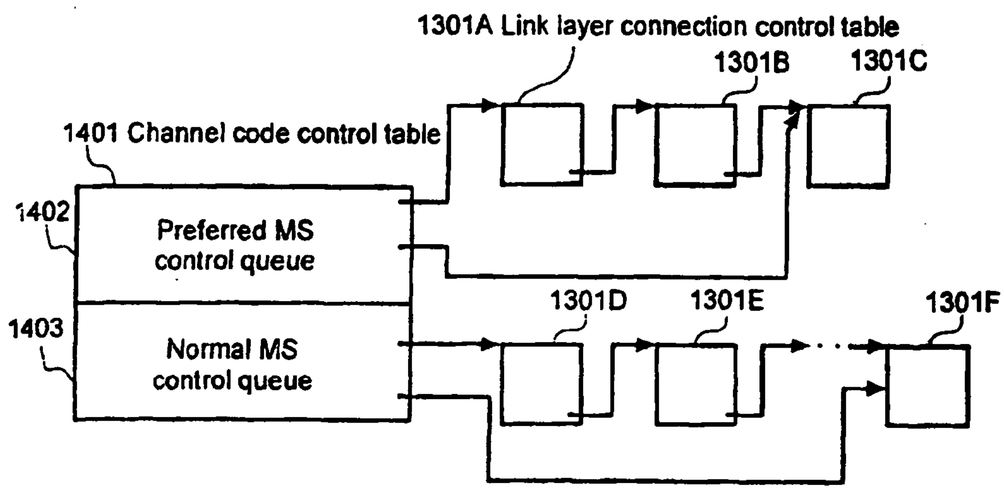

1301A Link layer connection control table
1301B
1301C
1401 Channel code control table
1402
Preferred MS control queue
1403
Normal MS control queue
1301D
1301E
1301F

FIG.15

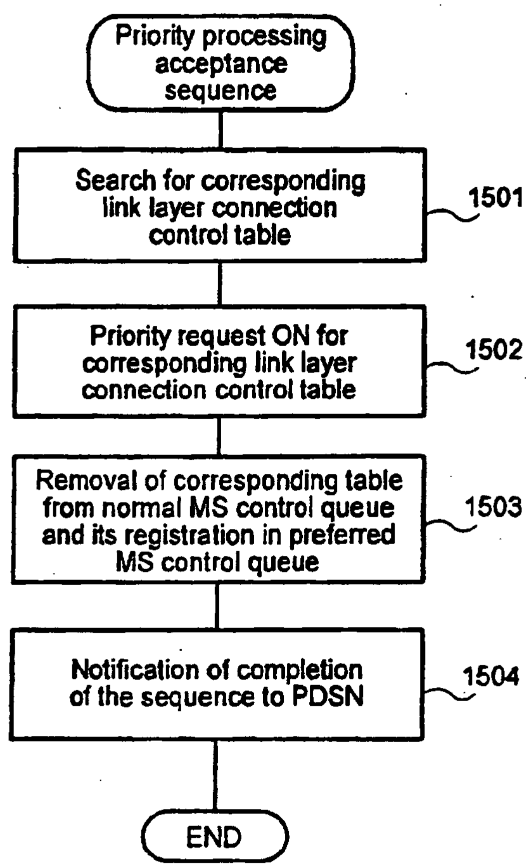

Priority processing acceptance sequence
Search for corresponding link layer connection control table
1501
Priority request ON for corresponding link layer connection control table
1502
Removal of corresponding table from normal MS control queue and its registration in preferred MS control queue
1503
Notification of completion of the sequence to PDSN
1504
END

FIG.16

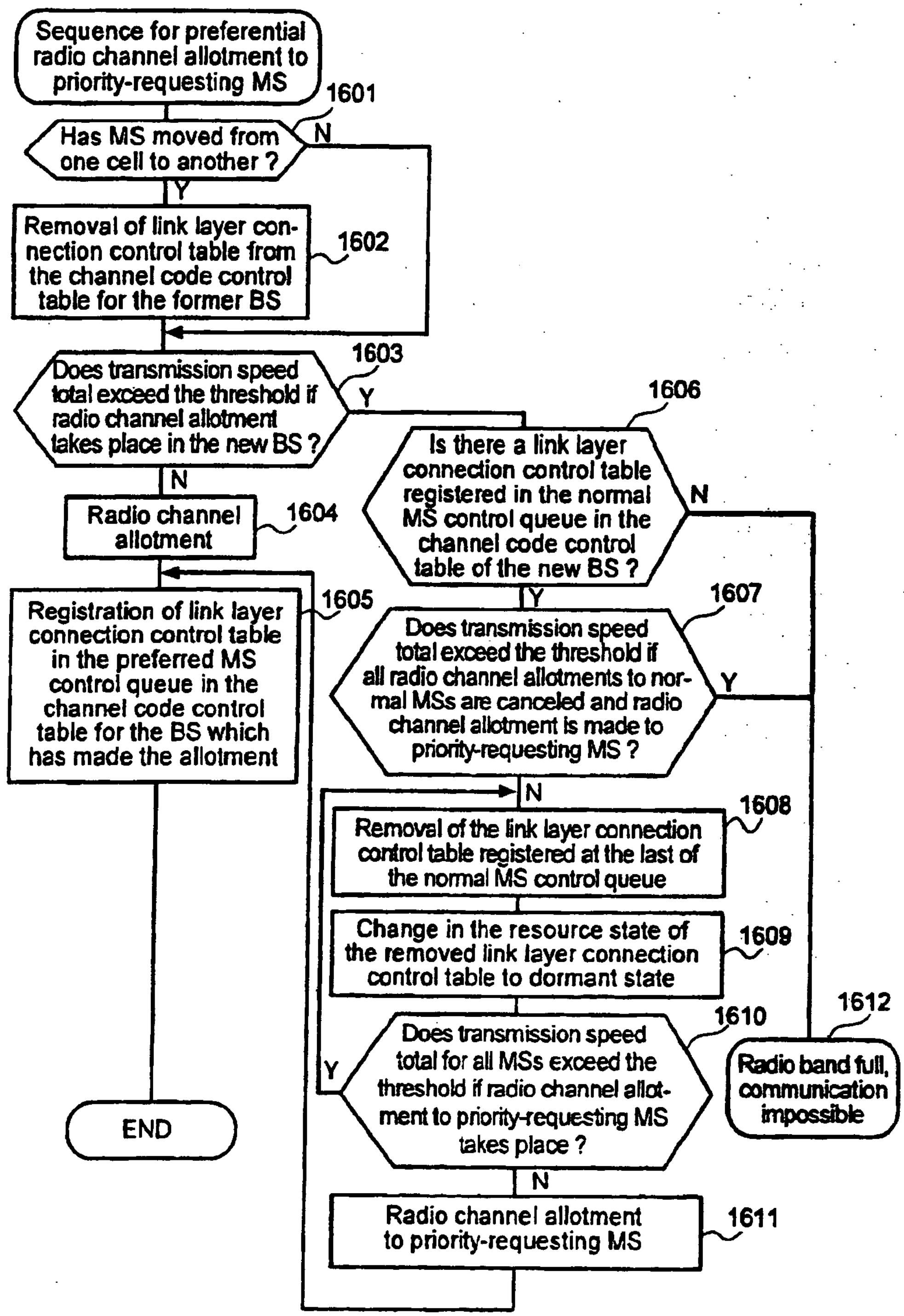
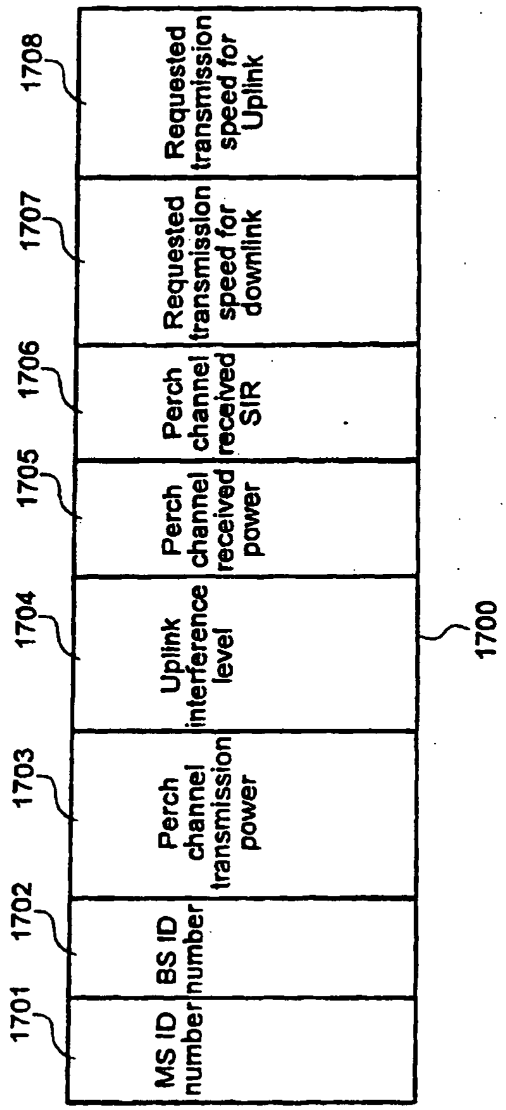

Sequence for preferential radio channel allotment to priority-requesting MS
1601
Has MS moved from one cell to another ?
N
Y
Removal of link layer connection control table from the channel code control table for the former BS
1602
1603
Does transmission speed total exceed the threshold if radio channel allotment takes place in the new BS ?
Y
N
Radio channel allotment
1604
Registration of link layer connection control table in the preferred MS control queue in the channel code control table for the BS which has made the allotment
1605
END
1606
Is there a link layer connection control table registered in the normal MS control queue in the channel code control table of the new BS ?
N
Y
1607
Does transmission speed total exceed the threshold if all radio channel allotments to normal MSs are canceled and radio channel allotment is made to priority-requesting MS ?
Y
N
Removal of the link layer connection control table registered at the last of the normal MS control queue
1608
Change in the resource state of the removed link layer connection control table to dormant state
1609
1610
Does transmission speed total for all MSs exceed the threshold if radio channel allotment to priority-requesting MS takes place ?
Y
N
1612
Radio band full, communication impossible
Radio channel allotment to priority-requesting MS
1611

FIG.17



1701
MS ID number
1702
BS ID number
1703
Perch channel transmission power
1704
Uplink interference level
1705
Perch channel received power
1706
Perch channel received SIR
1707
Requested transmission speed for downlink
1708
Requested transmission speed for Uplink
1700

FIG.18

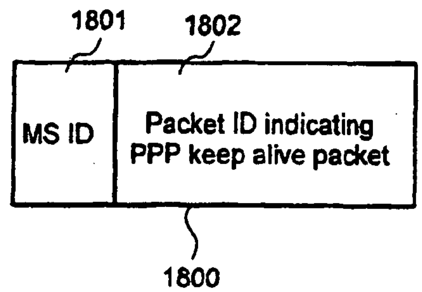


FIG.19

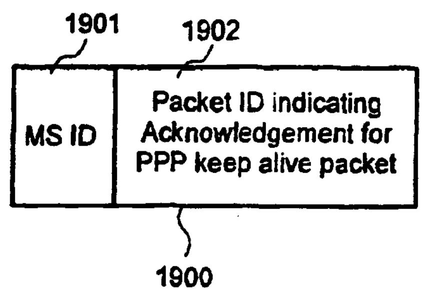
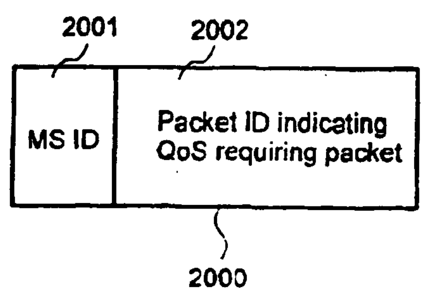


FIG.20



Design, Implementation, and Evaluation of Highly Available Distributed Call Processing Systems

Kazutaka Murakami, Richard W. Buskens, Ramachandran Ramjee, Yow-Jian Lin, Thomas F. LaPorta
Bell Laboratories, Lucent Technologies
101 Crawfords Corner Road
Holmdel, NJ 07733-3030

Abstract

This paper presents the design of a highly available distributed call processing system and its implementation on a local area network of commercial, off-the-shelf workstations. A major challenge of using off-the-shelf components is meeting the strict performance and availability requirements in place for existing public telecommunications systems in a cost-effective manner. Traditional checkpointing and message logging schemes for general distributed applications are not directly applicable since call processing applications built using these schemes suffer from high failure-free overhead and long recovery delays. We propose an application-level fault-tolerance scheme that takes advantage of general properties of distributed call processing systems to avoid message logging and to limit checkpointing overhead. The proposed scheme, applied to a call processing system for wireless networks, shows average call setup latencies of 180ms, failover times of less than three seconds, and recovery times of less than seventeen seconds. System availability is estimated to be 0.99995. The results indicate that using our proposed scheme meets the above challenge.

1 Introduction

Building call processing, or switching, systems in a distributed environment of general purpose computers is becoming the trend of next generation switching systems [2][10][17]. Emerging distributed call processing architectures illustrate that a properly modularized and functionally distributed software architecture increases system scalability, performance, and flexibility [9][10][17]. Moreover, advances in open distributed processing, where the Common Object Request Broker Architecture (CORBA) [13] is one example, facilitate portable and interoperable implementations of distributed software architectures in a heterogeneous computing environment. Systems that use these technologies are easily able to leverage off the continually increasing performance-to-price ratio of off-the-shelf computing components.

The stringent performance and availability requirements for public telecommunications systems pose key challenges to developing highly available distributed call processing systems that use off-the-shelf commodity hardware and software platforms. Call processing software must process each call request within a few hundred milliseconds [11], and the entire switching system should not be out of service for more than a few minutes per year [18]. Existing switching systems rely on custom-designed fault-tolerant processors and special-purpose operating systems to meet these requirements. Next generation switching systems built using general purpose computing platforms, however, require the development of software-based fault-tolerance solutions which achieve the same performance and availability goals.

Checkpointing and message logging strategies have been extensively discussed in the literature to enhance the level of software-based fault tolerance in a distributed computing environment [3] [14]. In general, a snapshot of the entire state of a software process is saved periodically (checkpointing), while messages sent or received by the process are logged (message logging) between checkpoints. Assuming a piecewise deterministic execution model, the state of the process can be reconstructed during recovery by replaying logged messages in their original order [5]. These techniques can be embedded into the operating system, with the advantage of being virtually transparent to application software [4].

For the reasons listed below, periodic checkpointing with message logging between checkpoints is not directly applicable to distributed call processing systems that use commodity components. First, large service delays due to periodic checkpointing of the entire state of a process will likely exceed call setup latency requirements during checkpointing. Second, the increased number of message exchanges required for distributed call processing makes it too time-consuming to log each message and still satisfy latency requirements. Third, periodic checkpoint intervals tend to be on the order of tens of minutes or more to limit the computational overhead of checkpointing global states;

hence the number of logged messages to be replayed after a failure is large, resulting in longer recovery time and lower system availability. In addition, a majority of these replayed messages are superfluous, since the expected lifetime of a call (around three minutes) is generally shorter than the checkpoint interval.

This paper presents the design, implementation, and evaluation of a high performance, highly available distributed call processing system based on commodity hardware and software platforms. Our primary contributions in this paper are: (1) the identification of general properties of distributed call processing systems; and (2) the development of an application-level fault-tolerance scheme that takes advantage of these properties to reduce checkpointing overhead, avoid message logging, and shorten recovery time. The scheme uses *object-level checkpointing*, defines a *selective event-driven checkpointing policy* where checkpoints are selectively taken at different servers when certain events occur, introduces *state reloading* as a technique to reload state in a recovering server based on redundant state information maintained by other servers, and conducts recovery procedures at relevant servers asynchronously. We demonstrate the effectiveness of our approach through measurements and analysis of our implementation of a call processing system for wireless networks. Although the above techniques have been developed for distributed call processing applications, these same techniques can also be applied to other classes of applications with similar properties (Section 3 covers this in detail).

The remainder of the paper is organized as follows. Section 2 introduces a reference model for distributed call processing. The model facilitates our discussion on highly available distributed call processing in Section 3, which also presents the details of our fault-tolerance scheme. This scheme is applied to a distributed call processing system for wireless networks. Section 4 discusses the system architecture and its implementation. In Section 5, a performance evaluation of our implementation measures call setup latency, failure-free fault-tolerance overhead, and recovery time. Section 6 estimates overall system availability, followed by concluding remarks in Section 7.

2 Reference model for distributed call processing

A telecommunications network architecture consists of many functional entities (FEs) that perform distinct tasks in the network. For example, the WIN Distributed Functional Plane defines a distributed functional model for wireless intelligent networks [15]. The model includes FEs that provide call control, access control, service control, and location registration functions. Call processing scenarios refer to various groupings of tasks coordinated through sequences of signaling messages. A distributed call processing system is a mapping of tasks to a collection of cooperating software modules. In general a software module could support tasks of multiple FEs, but only one software module is responsible for all tasks of a single FE. The modular design of functional models and its distributed implementation enable rapid introduction of new media and services over emerging transport technologies. It is also easier to scale up/down the capacity of a distributed call processing system and to enhance the capabilities of individual software modules.

We define four distributed call processing terms based on object-oriented concepts: two object classes, *functional object class* and *server class*, and two object instances, *functional object* and *server instance*. A functional object class corresponds to an FE. It defines unique call processing functions supported by the class, types of physical and logical resources managed by the class, and interfaces exported to other functional object classes. A functional object is an instance of a functional object class. Each functional object manages its own assigned resources and associated data corresponding to a single call activity. Multiple functional object classes may be needed to service a single call processing request. Each request results in the creation of one functional object for each of these functional object classes. Collectively, these functional objects maintain overall state information related to the request. The functional objects exist until the requested activity (e.g., a call) ends.

A server class corresponds to a software module. It is a unit of computation in realizing the abstract concept of a functional object class in a distributed call processing architecture. Server classes support one or more closely related functional object classes. A server instance is an embodiment of a server class, and typically corresponds to a process in a real implementation. To make its capacity scalable, a call processing system can have multiple instances of the same server class. Note that the term *server* is used to represent either a server class or a server instance; the meaning should be clear from the context.

As an example, Figure 1 depicts four classes of functional objects identified in our case study Mobile Switching Center (MSC) [9] presented in Section 4: *connections* (CONN), *channels* (CHAN), *calls* (CALL), and *user agents* (UA). A CONN object performs the necessary tasks for establishing a single end-to-end connection and maintains detailed state information about this connection. A CHAN object controls resource allocation activities for a specific transport channel resource, such as the channel of a switching device in the MSC. A CALL object records call activities of a specific user, while a UA object maintains non-call-related state information about the user (such as a user's service profile). Note that UA and CALL are user-specific, a CONN is unique for each connection,

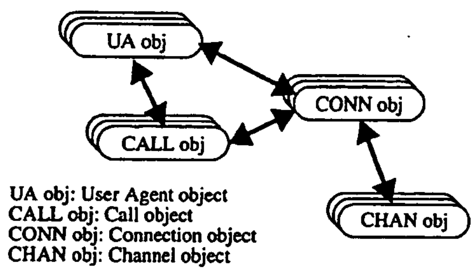


Figure 1. Functional objects for distributed call processing

and a CHAN is for a particular resource. Thus, UA and CALL object classes are good candidates for being grouped together within one server class.

3 Design for highly available distributed call processing systems

As stated in the introduction, an operating system level fault-tolerance approach has three major obstacles (global state checkpointing, frequent message logging, and long message replaying) to meeting the stringent performance and availability requirements for distributed call processing systems. Consequently, we use an application-level fault-tolerance solution [4][5]. This approach, although not transparent to application software, allows significantly finer control over when and what data to checkpoint, and how to perform recovery. Our goals are: (1) to perform checkpointing at a fine granularity to avoid long checkpointing times; and (2) to eliminate message logging, thereby reducing failure-free overhead and expediting recovery procedures. In this section, we describe our fault-tolerance scheme, which is derived by exploiting properties common to distributed call processing systems. After stating the requirements for call processing systems, we describe our checkpointing and recovery strategies.

3.1 Requirements

Public telecommunications switching systems have been designed to meet stringent availability requirements due to a large societal dependence on their service: only a few minutes of downtime per year are tolerated [18]. Since we cannot prevent failures from occurring, recovery times must be short to minimize service down time. In addition, any design of highly available distributed call processing systems must satisfy the following requirements:

R1. *High performance*: Low end-to-end call setup times (less than a few hundreds milliseconds) are required [11]. This constraint must be satisfied even with provisions for high availability.

R2. *Active call preservation*: Active calls must be preserved across failures. Calls in a transient state, on the other hand, need not be preserved, but may be retried or cleared. Clearing transient calls is a common practice in telecommunications systems.

R3. *Resource leak avoidance*: Reserved system resources must be released even if a call request is abnormally aborted due to a failure.

3.2 Object-level checkpointing

Our fault-tolerance scheme takes checkpoints per functional object instead of per process, which we call *object-level checkpointing*. The following general property of call processing supports this approach:

Property 1: Functional objects are independent and small in size.

Since a call activity involves only one functional object per functional object class, there is no mutual dependency among functional objects of the same class. Thus, checkpointing can be scheduled per object without coordinating with other objects in the class. Since call processing systems in public telecommunications networks can handle a large amount of call signaling traffic, a process may contain thousands of functional objects[1]. Each checkpoint thus contains only a tiny fraction of the entire process state.

Even if checkpoints are taken per object, message logging is still required to recover from lost messages. Due to the following property of call processing systems, however, we can completely eliminate message logging:

Property 2: Call processing systems are surrounded by robust standard signaling interfaces.

Switching systems of different vendors and in different networks must interwork through standard signaling interfaces. Signaling protocols for these interfaces have been designed with high reliability in mind; detecting lost request or response messages and invoking appropriate recovery actions are all part of the protocols. Timeout mechanisms are commonly used to detect failures. Upon timer expiration, a lost request is either retried or aborted, depending on the situation. Therefore, neither message logging nor message replay is necessary, resulting in lower failure-free overhead and reducing recovery time.

3.3 Checkpointing policy

An important design choice for object-level checkpointing is determining when to checkpoint a functional object. The most intuitive approach is to checkpoint the object whenever its state changes (due to a message receipt). Since many message exchanges are involved in a single call setup request, however, this method significantly degrades failure-free performance. A more suitable checkpointing policy is therefore needed.

1. For example, consider a local switch that can process 200K calls per hour. Assuming a 3-minute call holding time, on average ten thousand calls (and CONN/CALL objects) are active at a given time.

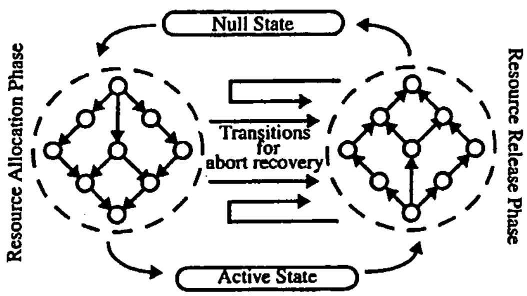


Figure 2. Typical state machine in call processing systems

Before developing our checkpointing policy, let us review the structure of typical call processing software to determine where and when to best perform checkpointing. A characteristic feature of call processing systems is the asynchronous nature of events. Since multiple parties are involved in a call, two independent, and sometimes conflicting, events may simultaneously affect a single functional object. For example, a caller might hang up while connections are being setup for the call. Upon arrival of such an interruptive event, it may be necessary to abort ongoing procedures for the original request.

To cope with asynchronous event arrivals, a state machine model is employed for telecommunications systems. Figure 2 shows a typical state machine for call processing systems. Two *stable* states, *null state* and *active state*, exist along with many other *transient* states. For the CONN object introduced in Section 2, for example, the active state represents the state where an entire connection is established between end users, while the null state means that there is no connection. The transient states for a CONN object are the states where the connection is being setup or torn down. The following observation of transient states supports our checkpointing policy:

Observation: Only a small number of calls are in a transient state.

Since call establishment and call release procedures take only a few hundred milliseconds compared to average call durations, which are on the order of minutes, most call activities are in stable states.

Based on the above properties of distributed call processing systems, we describe our checkpointing policy, called *selective event-driven checkpointing*. The policy omits as many checkpoints as possible, except where omitting a checkpoint violates two of the requirements discussed in Section 3.1, namely active call preservation (R2) and resource leak avoidance (R3). Checkpoints are taken only when:

1. committing to a stable state, and
2. obtaining new state information required to undo resource allocation or to redo resource clearing.

In our approach, all objects in a transient state within a failed server instance are cleared. Since most calls are in stable states, only a small number of calls are affected by the above checkpointing policies.

The following observation for distributed call processing systems makes it possible to further reduce the number of checkpoints:

Property 3: Partial state information is replicated among multiple objects of different functional object classes.

When functional objects are contained within different servers, replicated state information may exist among the servers so that a functional object in one server can identify an appropriate functional object in another server. We avoid redundantly checkpointing the same data by designating one server to be responsible for checkpointing the state information shared by the servers. After a failure, a recovering server that does not checkpoint the shared state 'reloads' its state information from servers that do checkpoint the state. We call this procedure *state reloading*. State reloading reduces the number of checkpoints in the system, reducing failure-free fault-tolerance overhead.

We distinguish *pessimistic state reloading* (PSR) from *optimistic state reloading* (OSR). In PSR, new call setup requests that arrive at a recovering server before state reloading completes are discarded. In OSR, new call setup requests that arrive at a recovering server are processed *while* state reloading is being performed, based on the assumption that call setup requests do not arrive for users that are already in a call. Thus, OSR decreases system recovery time as compared to PSR. Occasionally, the above assumption may not be satisfied, resulting in conflicting state information. When this occurs, ongoing call request procedures for the conflicting call must be aborted.

3.4 Call recovery strategy

During recovery, a recovering server instance must either undo or redo unsuccessful call setup and release attempts, detect state inconsistencies, and resynchronize the states of related objects among distributed servers. Since our fault-tolerance scheme is performed at the application level, these recovery mechanisms must also be realized at the application level. Incorporating these mechanisms for fault tolerance is easy since most of the required recovery procedures are already present in normal call processing flows.

Recall from Figure 2 that there are two main phases in processing call requests. The *resource allocation phase* reserves network resources in stages during the transition from a null state to an active state. The *resource release phase* returns the call processing state machine to a null state from an active state by freeing reserved resources. Additional state transitions exist from transient states in the resource allocation or resource release phase to states

in the resource release phase. They embody an abort action triggered by an interruptive event like a timeout or hang-up by a caller. Since such events may occur asynchronously with respect to the current state, call processing systems are required to provide *abort recovery procedures* for each functional object from any state. Furthermore, an interruptive event at one server may cause inconsistencies among the states of related functional objects in different servers. Thus, distributed call processing software must provide a *global resynchronization procedure* to resynchronize the states of the related objects. Abort messages which initiate abort recovery procedures for a functional object may be used for this purpose. Due to the asynchronous arrival of such events, the precise state of an interrupted resource reservation request, for example, is unknown, and it is uncertain if the request is granted or not. Therefore, abort recovery operations must be *idempotent*, namely, carrying them out several times has the same effect as carrying them out once [12]. To summarize:

Property 4: Distributed call processing systems furnish idempotent operations, abort recovery procedures, and global resynchronization procedures.

Given Property 4, only minimal effort is required to support recovery from failures. To avoid resource leaks, a recovering server instance must initiate abort recovery procedures for functional objects that are in transient states and invoke system-wide resynchronization procedures as necessary. The idempotent resource release operations permit fewer checkpoints to be taken during call setup and call release, with no adverse effects of unnecessarily reissuing release requests during recovery.

3.5 Primary and backup configuration

To further shorten recovery time after a failure, a primary-backup approach is used for each server instance [8], as opposed an approach that checkpoints state information to disk. To survive a single host failure, the primary and its backup execute on different hosts. The primary server instance processes all incoming requests and checkpoints state information to its backup, as necessary. Since a backup server is already executing when a primary failure occurs, server unavailability is reduced due to shorter failover times.

4 Case study: Mobile Switching Center

This section applies the ideas of functionally distributed call processing and the general techniques for high availability, as described in the previous two sections, to a call processing system for wireless networks, called a *Mobile Switching Center* (MSC) [9]. After providing background information on mobile switching centers, we present our MSC architecture, the checkpointing and reloading strategies, and the failure detection and recovery mechanisms.

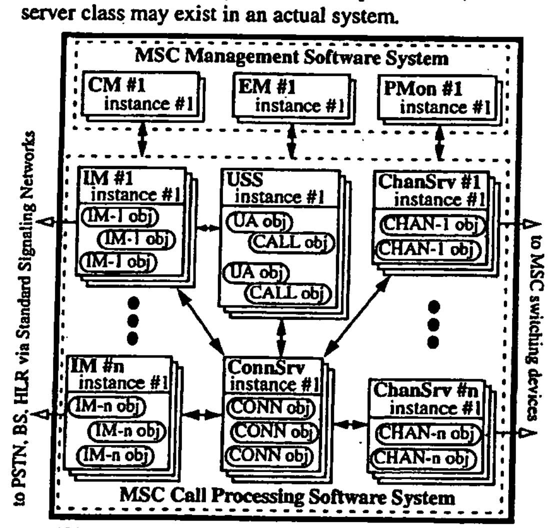
4.1 Background

An MSC is a local switching facility in a wireless network. Each MSC controls mobile traffic in a service area divided into geographical regions called *cells*. A *Base Station* (BS) within each cell manages radio resources between itself and all *Mobile Stations* (MS) roaming within the cell. All BSs in an MSC's service area have wired links to the MSC, which in turn connects to other MSCs and to the *Public Switched Telephone Network* (PSTN). A *Home Location Register* (HLR) is connected to the PSTN and keeps a global database identifying which MSC is responsible for setting up calls to a particular MS. The procedure of locating an MS within an MSC's service area during call setup is called *paging*.

An MSC performs two key functions: call processing and mobility management. Call processing involves setting up and tearing down connections between calling and called parties. Mobility management includes power-up and power-down registration of MSs, resulting in updates to the MS's location information in the corresponding HLR, and paging mobile stations during call setup.

4.2 Highly available MSC architecture

Figure 3 illustrates our MSC architecture. There are four classes of call processing servers and three classes of management servers. Note that multiple instances of each server class may exist in an actual system.



USS: User Signaling Server
ConnSrv: Connection Server
CM: Configuration Manager
PMon: Process Monitor
ChanSrv: Channel Server
IM: Interworking Manager
EM: Event Manager

Figure 3. MSC call processing software structure

4.2.1 **Call processing servers.** *Interworking managers* (IMs) act as protocol gateways to internal MSC servers,

isolating them from external signaling protocols and thereby allowing the MSC to evolve independent of these protocols. An IM may terminate one or more signaling protocols and one or more types of IMs may exist within an MSC. Functional objects within an IM record mapping information between identifiers, such as call id, used both internal and external to the MSC to correlate call processing activities.

A *user signaling server* (USS) maintains information in UA objects about the registration status of mobile stations currently roaming within the service area of the MSC. A USS also houses CALL objects, each recording call activities involving a particular mobile station.

Channel servers (ChanSrvs) maintain CHAN objects to manage switching device resources allocated during call setup and deallocated during call release. Examples of resources include the switching fabric used to setup physical connection segments and voice encoders/decoders that take packet data from the wireless link (air interface) and convert it to pulse code modulated voice and vice versa.

A *connection server* (ConnSrv) coordinates allocation of channel resources to setup a connection to the BS of the cell in which the MS is currently roaming. The ConnSrv instructs appropriate ChanSrvs to reserve needed MSC channel resources and sends messages to external components via IMs to reserve channel resources external to the MSC. Each ConnSrv maintains detailed state information about a single connection for an MS in a CONN object.

4.2.2 **Management servers.** A *Process Monitor* (PMon) detects failures of both server instances and processors. An *Event Manager* (EM) collects failure reports from PMons, performs fault isolation, and informs a *Configuration Manager* (CM) of actual failures. The CM coordinates appropriate system-wide recovery actions, including necessary reconfiguration activities; it also performs overall system initialization.

4.3 Checkpointing and state reloading strategies

The various call processing servers described above use different strategies for checkpointing and state reloading. ConnSrvs perform selective event-driven checkpointing of CONN objects using the checkpoint policy described in Section 3. Since all ConnSrv state is contained within CONN objects, state reloading is not needed. USSs perform selective event-driven checkpointing of UA objects and OSR for CALL objects. CALL objects can be derived from corresponding UA and CONN objects. PSR is used for CHAN objects to ensure that channel resources allocated before a ChanSrv failure are not mistakenly reallocated during recovery. CHAN objects can be recreated from information in CONN objects. The IMs in our architecture are stateless and therefore require no checkpointing or state reloading.

4.4 Failure detection and recovery

Process crashes and hangs are two major causes of *process failures* [4]. The former can be immediately detected by a PMon as an underlying connection break (e.g. TCP/IP connection) with the failed process, typically within a hundred milliseconds. Detecting process hangs is achieved by PMons periodically exchanging "keep alive" messages with each process. An unsuccessful "keep alive" message exchange indicates a potential failure of the process. We refer to this periodic message exchange interval as the *keep-alive interval*. The keep-alive interval determines failure detection time. We assume that PMons are very reliable and so do not fail unless the underlying processor fails. Then, detecting *processor failures* only requires us to detect PMon failures. To achieve this, PMons are deployed on all host machines and monitor each other using a (dynamic) testing assignment as described in [1][4].

Once a failure is detected, recovery actions are initiated. The following list enumerates the recovery steps that occur following the hang of a primary server instance (process) that uses PSR:

1. The PMon reports the unsuccessful "keep alive" message exchange to EM;
2. EM performs fault isolation to identify the server instance that has failed;
3. This failure is reported to the CM, which coordinates all remaining recovery actions;
4. Signaling connections are established between the failed server instance's backup and all server instances originally communicating with the failed server instance;
5. State reloading procedures are initiated in the backup server instance, if necessary;
6. Once state reloading is complete, the backup server instance becomes a primary server instance. This new primary commences state resynchronization procedures and starts accepting new incoming call processing messages. Call processing messages that arrive before this step are discarded;
7. After the new primary becomes available, a new backup server instance is instantiated;
8. The new primary checkpoints its entire state to the new backup. This procedure is referred to as *checkpoint dumping*.

For recovery after backup server instance failures, steps 1-3, 7 and 8 are executed. Recovery actions initiated by the failure of a primary server instance that uses OSR take the same steps, except that the backup becomes a primary after step 4 and incoming calls that arrive during state reloading are processed instead of discarded.

4.5 Implementation

An implementation of our MSC architecture executes on a collection of UNIX workstations interconnected via a local area network. Each MSC call processing server is implemented as a UNIX process. Inter-process communication between MSC servers uses Iona's Orbix [6], a CORBA-based middleware platform. Server instances are implemented as CORBA objects, while functional objects are realized as C++ objects. The MSC implementation includes three classes of IMs to support standard telecommunications signaling interfaces: an IS-634A interface with base stations [16]; an IS-41 interface with an HLR [15]; and an ISDN User Part interface with PSTN switching nodes [7]. A single class of channel servers executes on an embedded system that provides frame selection and voice encoding/decoding capabilities.

In our implementation, a mobile station registration scenario involves four CORBA message exchanges within MSC servers and a single checkpoint when the registration state (powered up or powered down) changes. Processing a call setup request originated from a mobile station, i.e. a *call origination* scenario, involves nine CORBA message exchanges and three checkpoints, while a call setup request coming from PSTN, i.e. a *call termination* scenario, requires seventeen CORBA message exchanges and two checkpoints. A call release request involves nine CORBA message exchanges and two checkpoints. Note that our proposed scheme requires only 25% of all state transitions due to message arrivals to be checkpointed, considerably reducing failure-free overhead compared to traditional approaches.

5 Performance evaluation

This section presents the results of a performance evaluation of our MSC implementation. Failure-free overhead and recovery times after failures are discussed.

5.1 Experimental environment

The MSC hardware platform used in our experimentation consists of two SUN Ultra 2 workstations, each housing a single 200 MHz UltraSPARC-I processor, interconnected via a 10 Mbps Ethernet. The MSC software configuration consists of two instances each of the USSs and ConnSrvs, one PMon instance per workstation, and one instance each of the other MSC call processing and management servers. We distribute all server instances, including backups for each primary call processing server, across the two workstations. Two simulators are employed to generate user registration and call processing traffic, one to simulate a BS and the other to simulate the HLR and PSTN switching nodes. The simulators execute on separate UltraSparc workstations and exchange call processing

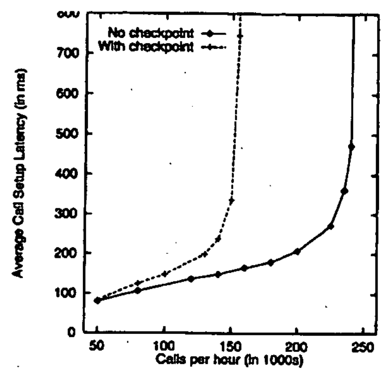


Figure 4. Failure-free overhead: latency versus load with and without checkpointing

messages with the MSC via TCP/IP connections to the IMs. In our experiments, a Poisson distribution models registration and call request arrival traffic. An exponential distribution models call holding time.

5.2 Failure-free overhead

To evaluate the failure-free checkpointing overhead of the MSC, we measured call setup latencies of our experimental configuration both with and without checkpointing to backup servers. We assume 40,000 registered mobile stations and a 90-second average call holding time. Performance with several call arrival rates are examined, while the ratio of originating to terminating calls is kept fixed at a ratio of approximately two-to-one. Power-up registration for all mobile stations is performed in advance of the performance evaluation, although registration traffic for handoffs does take place during the measurement period. This handoff registration traffic does not incur any checkpointing at the USSs. In the experiments, call setup latency is measured at the two simulators since this reflects the delay incurred within the MSC as perceived by end users.

Figure 4 shows the average call setup latency versus call load. The curve has a knee when the latency goes beyond 180 milliseconds. At the knee, call throughput is 120K calls/hour with fault-tolerance support (180K calls/hour if checkpointing is not performed). This represents the maximum call throughput for the given system configuration; beyond this call throughput, the MSC becomes overloaded. Note that checkpointing to backup processes reduces call throughput by 33%.

5.3 Recovery time

Table 1 summarizes the mean recovery time at 120K calls/hour for crash failures of the primary instances of the various MSC servers. For illustration purposes, recovery times for primary USS failures are presented both for PSR and OSR. Forty samples are collected for each case. Timestamps were taken by the CM at four different events

Failed Primary Server	USS w/ PSR	USS w/ OSR	Conn Srv	IM	Chan Srv
ReconnDoneTime (s)	0.29	0.27	0.16	0.15	0.20
ReloadDoneTime (s)	1.73	2.13	0.20	0.18	2.53
DumpChkptTime (s)	5.10	5.95	4.09	—	—
DumpDoneTime (s)	16.33	16.48	5.15	—	—

TABLE 1. Recovery Time at 120K Calls/Hour

during recovery, relative to the time when the failure is first reported to the CM. These events are described below. Note that worst case recovery times are within 5% of the mean values indicated in Table 1.

ReconnDoneTime identifies the time when all lost signaling connections between the failed primary and other server instances are re-established with the backup. For ConnSrvs and USSs operating with OSR, the backup server instance is activated (becomes the new primary) at this point, and new call requests can be accepted. *ReloadDoneTime* identifies when state reloading is complete. For USSs with PSR, new call processing messages can be accepted after this point. For ConnSrv failures, the time between *ReconnDoneTime* and *ReloadDoneTime* is used to initiate resynchronization procedures for transient objects in the failed server. Due to the small number of transient calls in the system, this time difference is small. For USS failures, the difference between *ReconnDoneTime* and *ReloadDoneTime* is greater for OSR than for PSR since, for OSR, new call requests are accepted and processed during state reloading of CALL objects from the ConnSrvs. Our experimentation shows that OSR results in a 75% reduction of lost calls over PSR.

Approximately 3.5 seconds elapse between *ReloadDoneTime* and *DumpChkptTime* for the CM to create a new backup process. *DumpChkptTime* identifies the time at which the primary performs checkpoint dumping to store a copy of its state information at the new backup, and *DumpDoneTime* indicates when this procedure has completed. At a load of 120K calls/hour using a 90-second call holding time, each USS houses approximately 20,000 UA objects which must be downloaded during this time (corresponding to 2MB of data), while a ConnSrv contains only 1,500 CONN objects. This explains why roughly ten seconds are needed to checkpoint complete USS state information, compared to just over one second for ConnSrv failures. After *DumpDoneTime*, i.e. 5 to 17 seconds after the failure is reported to CM, the system is ready for the next failure of this specific server instance.

The short recovery times observed in our experimentation contribute to the high overall system availability derived in the following section.

6 Availability analysis

This section presents a preliminary availability analysis for our MSC prototype. We calculate MSC availability based on the number of lost calls, rather than the time that components of the MSC are unavailable, since lost calls represent a more useful metric of lost revenue in telecommunications call processing systems. The same MSC system configuration used in the performance evaluation (Section 5) is employed in this analysis.

USS recovery uses OSR, while ChanSrv recovery uses PSR. Call processing load at a particular type of server is assumed to be evenly distributed among all instances of the server. Although call arrival rates vary throughout the day, we take a conservative assumption that failures occur only during busy hours. We further assume that sufficient processing capacity exists within the MSC so that overall performance can be maintained during failover.

In our analysis, the following assumptions are made about the failure model. These assumptions allow us to determine MSC availability based on the experimental data presented in Section 5.3.

- Process hangs, operating system hangs, and processor crashes are considered[2].
- Communication device failures are not considered.
- Failures are independent.
- Multiple simultaneous failures do not occur..
- Recovery completes before the next failure occurs[3].

Furthermore, the following conservative assumptions on failure rates are used in our availability calculation:

- Each process fails (independently) once every other week.
- The operating system on each processor hangs once every two months.
- Each processor fails once every 18 months.

6.1 Lost calls resulting from process failures

There are three sources of lost calls resulting from failures: transient lost calls, discarded calls, and abandoned calls. *Transient lost calls* are the calls that are being setup when a process failure occurs. After the failure but before a new primary process becomes ready, all newly arrived calls are discarded; we refer to these as *discarded calls*. Finally, in our implementation, checkpointing all state information from a primary process to a new backup pro-

2. Our analysis assumes that process hangs are the only source of process failures. Since it takes a considerably longer time to detect a process hang than a process crash, our availability analysis will give us a more conservative result. Processor crashes require a similar fault detection period.

3. For processor failures, recovery does not imply that the failed processor is repaired before the next process failure occurs. It is assumed, however, that failed processors do get repaired and are reintegrated into the system (without disrupting call processing) before a subsequent operating system or processor failure.

cess during recovery prevents the primary process from accepting new requests. These new requests are not discarded but blocked at the input message queue of the primary process for the duration of checkpoint dumping. If call setup is not performed sufficiently quickly, a user may abort the call request. We refer to these calls as *abandoned calls* and capture this phenomenon in our analysis by a patience probability function $p(t)$, defined as the probability that a user hangs up a call at time t. In this analysis, we assume that a user always abandons a call if the call does not complete within a threshold time t_{TH} from the corresponding call request. Thus, the patience probability function $p(t)$ equals the delta function $\delta(t_{TH})$ such that $\int_0^t \delta(t_{TH}) = 1$ if $t \geq t_{TH}$, and 0 otherwise.

Let λ_{MSC} and λ_{s_i} denote the average call arrival rates at the MSC and at a primary server process s_i of server class s, respectively. If n_s represents the total number of instances of servers of class s, then $\lambda_{s_i} = \lambda_{MSC}/n_s$. Let t_{DT} be the time interval from a failure to its detection. For process hangs, t_{DT} is one half of a keep-alive interval on average. Furthermore, let t_{LT}, t_{RC}, t_{RL}, and t_{DM} denote average values of call setup latency, *ReconnDoneTime*, time required for state reloading (*ReloadDoneTime* - *ReconnDoneTime*), and time required to dump checkpoint data (*DumpDoneTime* - *DumpChkptTime*), respectively. Then, the total number of lost calls due to failure of a primary server process s_i, LC_{s_i}, is given by:

$$LC_{s_i} = \lambda_{s_i} t_{LT} + \lambda_{s_i}(t_{DT} + t_{RC} + \delta_{PSR} t_{RL}) + \lambda_{s_i} \int_0^{t_{DM}} p(t)dt \quad \text{(EQ 1)}$$

where the indicator variable δ_{PSR} equals 1 (0) if PSR (OSR) is used. The first term in Equation 1 corresponds to the number of transient lost calls, the second term represents the number of discarded calls, and the third term is the number of abandoned calls. When calculating the number of transient lost calls, a call is considered transient until call setup procedures have completed within the MSC. Call discarding occurs at three time intervals during process failure and recovery: before the failure is detected, after failure detection but before inter-process connections are re-established, and, if PSR is used, while state reloading is in progress. The terms in parenthesis in the second term of Equation 1 correspond, respectively, to these three time intervals. Note that only abandoned calls contribute to lost calls when backup processes fail.

Table 2 presents the number of lost calls due to failures of the primary and backup server processes at λ_{MSC} =120K calls/hour. We obtain t_{LT} = 180 milliseconds from Figure 4 and use a keep-alive interval of ten seconds (average process failure detection time t_{DT} = 5 seconds). t_{RC}, t_{RL}, and t_{DM} are taken from Table 1. We also use t_{TH} = 5 seconds.

The table indicates that the two major causes of lost calls are discarded calls due to long failure detection times and abandoned calls. Effects of the former can be reduced by shortening the keep-alive interval, with a resulting increase in keep-alive message traffic and a higher possibility of misinterpreting traffic congestion as a failure. Increasing the number of primary instances of a server is another approach, since distributing the call request load among more instances implies that each instance services, on average, fewer calls. For example, with twice as many USSs as IMs in the configuration, failure detection time contributes half as many lost calls for a USS failure as for an IM failure, as shown in the $\lambda_{s_i} t_{DT}$ column of Table 2.

MSC Server	n_s	Trans-ient lost calls	Discarded calls: $\lambda_{s_i} t_{DT}$	Discarded calls: $\lambda_{s_i} t_{RC}$	Discarded calls: $\delta_{PR} \lambda_{s_i} t_{RL}$	Aban-doned calls	Total lost calls
USS_{prim}	2	3.00	83.33	4.50	0.00	92.17	183.00
$Conn_{prim}$	2	3.00	83.33	2.67	0.00	0.00	89.00
IM_{prim}	1	6.00	166.67	5.00	0.00	0.00	177.67
$ChanSrv_{prim}$	1	6.00	166.67	6.67	84.33	0.00	263.67
USS_{back}	2	--	--	--	--	92.17	92.17
$Other_{back}$	1/2	--	--	--	--	0.00	0.00
Average	--	2.25	62.50	2.25	5.27	23.04	95.31

TABLE 2. Lost Calls Per MSC Server Process Failure

Abandoned calls resulting from USS failures are due to the long checkpoint dumping time needed to transfer UA object state from a primary USS to its backup. This time can be reduced by increasing the number of USS instances so that each USS instance manages fewer UA objects.

6.2 Lost calls due to operating system and processor failures

Both operating system and processor failures mean that all processes executing on the failed platform must be migrated to another processor, with primary-to-backup failovers occurring as necessary to speed up recovery time. The difficulty in calculating the number of lost calls is in identifying which processes must be failed over as well as the failover duration. Thus, we take the following conservative approach, assuming that recovery can proceed in parallel for each failed server instance: let S be the set of all server classes in the MSC and let $n = \min(n_s | s \in S)$. Thus, one n-th of transient calls and one n-th of all newly arriving calls will be affected until the longest recovery sequence completes, namely from the failure to *DumpDoneTime*. Let t_{DD} be the largest *DumpDoneTime*. Then, the total number of calls lost during failover due to operating system and hardware failures, LC_{OH}, is estimated by:

$$LC_{OH} = \lambda_{MSC}(t_{LT} + t_{DT} + t_{DD})/n \quad \text{(EQ 2)}$$

At λ_{MSC} =120 K calls/hour, t_{LT} = 180 milliseconds (from Figure 4) and t_{DD} = 16.48 seconds (from Table 1). Setting n = 1 and using t_{DT} = 5 seconds, we obtain LC_{OH} = 722 lost calls.

6.3 Availability calculation

Table 3 summarizes the number of lost calls per year due to various MSC component failures by using the failure rate assumptions noted at the beginning of Section 6. The availability is then easily calculated as (1 - total lost calls / attempted calls). Assuming a call arrival rate of 120K calls per hour, each and every day, 52 weeks a year, the analysis yields an MSC availability of 0.99995. This is in line with existing requirements for MSC availability.

MSC Component	Failures per Component per Year	Component Quantity	Lost Calls per Failure	Total Lost Calls/ Year
MSC Server Process	26.00	16	95.31	39648.96
Operating System	6.00	2	722.00	8664.00
Processor	0.67	2	722.00	967.48
TOTAL				49280.44

TABLE 3. Number of Lost Calls Per Year Based on MSC Component Failure

7 Conclusion

This paper presented the design, implementation, and evaluation of a highly available distributed call processing system based on commercial computing platforms. Call processing software is partitioned into functionally distinct servers, and software-based fault tolerance solutions are used exclusively to provide high availability. The use of object-level checkpointing, a selective event-driven checkpointing policy, and state reloading addresses the key challenge of meeting stringent availability and performance requirements simultaneously. Our scheme leverages off properties common to distributed call processing systems to reduce checkpointing overhead, avoid message logging, and shorten recovery time. It also requires less recovery coordination for resynchronizing states of relevant servers.

The proposed scheme was applied to a call processing system for wireless networks and implemented for operation using a local area network of UNIX workstations and CORBA as the communication middleware. The implementation, based on only two UltraSPARC-I processors connected via a 10 Mbps Ethernet, achieves a call handling capacity of 120K calls/hour with an average call setup delay of 180 milliseconds. Only one-fourth of all system-wide state changes are checkpointed using our selective event-driven checkpointing policy, with no resource leakage when failures occur. Failure-free overhead was measured to be 33%. Failover is fast, occurring within three seconds for single process failures, while total system recovery completes within seventeen seconds. Using pessimistic assumptions about the failure rates of software processes, operating systems, and processors, overall system availability is estimated to be 0.99995. This work demonstrates that it is feasible to build distributed call processing systems using off-the-shelf components that meet the stringent performance and availability requirements in place for existing custom-built systems.

References

[1] R. Bianchini and R. Buskens, "Implementation of On-Line Distributed System-Level Diagnosis Theory," *IEEE Transactions on Computers*, vol.41, no.5, pp.616-626, May 1992.

[2] H. Blair, S.J. Caughey, H. Green and S.K. Shrivastava, "Structuring Call Control Software Using Distributed Objects," *International Workshop on Trends in Distributed Computing*, Aachen, Germany, 1996.

[3] E.N. Elnozahy, D.B. Johnson and Y.M. Wang, "A Survey of Rollback-Recovery Protocols in Message-Passing Systems," Tech. Report CMU-CS-96-181, School of Computer Science, Carnegie Mellon University, Oct. 1996.

[4] Y. Huang and C. Kintala, "Software Fault Tolerance in the Application Layer," In *Software Fault Tolerance* (M.R.Lyu, Ed.), John Wiley & Sons, Chichester, England, pp.231-248, 1995.

[5] Y. Huang and Y-M. Wang, "Why Optimistic Message Logging has not been used in Telecommunications Systems," *Proceedings of the 25th International Symposium on Fault-Tolerant Computing*, pp. 459-463, 1995.

[6] IONA, Orbix 2: Reference Guide, Nov., 1995.

[7] ITU-T Recommendations Q.761-767, "The ISDN User Part of Signalling System No. 7", 1993.

[8] P. Jalote. "Fault Tolerance in Distributed Systems". Prentice Hall, 1994.

[9] T.F. LaPorta, A. Sawkar and W. Strom, "The Role of New Technologies in Wireless Access Network Evolution," in *Proceedings of International Switching Symposium (ISS '97)*, IS-03.18, 1997.

[10] T.F. LaPorta, M. Veeraraghavan, P.A. Treventi and R. Ramjee, "Distributed Call Processing for Personal Communication Services," *IEEE Communications Magazine*, vol.33, no.6, pp. 66-75, Jun. 1995.

[11] A.R. Modarressi, R.A. Skoog, "Signaling System No. 7: A Tutorial", IEEE Communications Magazine, Vol. 28, No. 7, pp. 19-35, July 1990.

[12] S. Mullender, "Distributed Systems," 2nd ed., ACM Press, New York, N.Y., 1993.

[13] Object Management Group (OMG), The Common Object Request Broker: Architecture and Specification, Rev. 2.0, July 1995.

[14] R.E. Strom and S. Yemini, "Optimistic Recovery in Distributed Systems," *ACM Transactions on Computer Systems*, vol.3, no.3, pp.204-226, Aug. 1985.

[15] TIA/EIA IS-41 (revision D): "Cellular Radio-Telecommunications Intersystem Operations," 1997.

[16] TIA/EIA IS-634 (revision A), 2nd ballot version, Oct. 1997.

[17] TINA-C, Service Architecture Version 2.0, March 1995.

[18] G. Zorpette, "Keeping the Phone Lines Open," *IEEE Spectrum*, vol. 26, no. 6, pp.32-36, Jun. 1989.